Noname manuscript No.
(will be inserted by the editor)

Fast RGB-D People Tracking for Service Robots

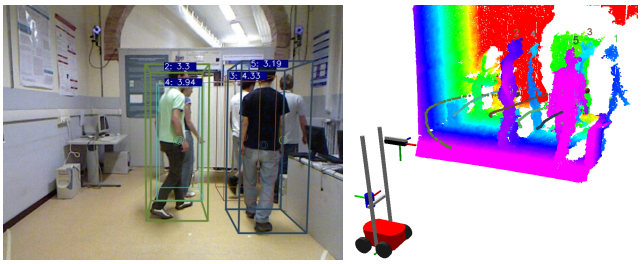
Matteo Munaro · Emanuele Menegatti

Received: date / Accepted: date

Abstract Service robots have to robustly follow and interact with humans. In this paper, we propose a very fast multi-people tracking algorithm designed to be applied on mobile service robots. Our approach exploits RGB-D data and can run in real-time at very high frame rate on a standard laptop without the need for a GPU implementation. It also features a novel depth-based sub-clustering method which allows to detect people within groups or even standing near walls. Moreover, for limiting drifts and track ID switches, an online learning appearance classifier is proposed featuring a three-term joint likelihood.
We compared the performances of our system with a number of state-of-the-art tracking algorithms on two public datasets acquired with three static Kinects and a moving stereo pair, respectively. In order to validate the 3D accuracy of our system, we created a new dataset in which RGB-D data are acquired by a moving robot. We made publicly available this dataset which is not only annotated by hand, but the ground-truth position of people and robot are acquired with a motion capture system in order to evaluate tracking accuracy and precision in 3D coordinates. Results of experiments on these datasets are presented, showing that, even without the need for a GPU, our approach achieves state-of-the-art accuracy and superior speed.

Matteo Munaro
Via Gradenigo 6A, 35131 - Padova, Italy
Tel.: +39-049-8277831
E-mail: matteo.munaro@dei.unipd.it

Emanuele Menegatti
Via Gradenigo 6A, 35131 - Padova, Italy
Tel.: +39-049-8277651
E-mail: emg@dei.unipd.it



(a) (b)

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 the estimated people trajectories.

Keywords People tracking, Service Robots, RGB-D, Kinect Tracking Precision Dataset, Microsoft Kinect.

1 Introduction and related work

People detection and tracking are among the most important perception tasks for an autonomous mobile robot acting in populated environments. Such a robot must be able to dynamically perceive the world, distinguish people from other objects in the environment, 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 monocular images only ([1], [2]) or range data only ([3], [4], [5], [6], [7]). 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 [8], 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 [9]

the authors propose a probabilistic aggregation scheme for fusing data coming from an omnidirectional camera, a laser range finder and a sonar system. These works, however, do not exploit sensors which can precisely estimate the whole 3D structure of a scene. 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 from a synchronized camera pair. The depth estimation provided by the stereo pair allowed them to reach good results in challenging scenarios, but their approach is limited by 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 ([12], [13]), but the computations needed for creating the disparity map always impose limitations to the maximum frame rate achievable, especially when further algorithms have to run in the same CPU. Moreover, they do not usually provide a dense representation and fail to estimate depth in low-textured scenes.

With the advent of reliable and affordable RGB-D sensors, we have witnessed a rapid boosting of robots capabilities. Microsoft Kinect sensor [14] allows to natively capture RGB and depth information at good resolution and frame rate. Even though the depth estimation becomes very poor over eight meters and this technology cannot be used outdoors, it constitutes a very rich source of information for a mobile platform. Moreover, Dinast [15] and PMD [16] recently built new depth sensors which can work outdoors, while Samsung created a CMOS sensor capable of simultaneous color and range image capture [17], thus paving the way for a further diffusion of RGB-D sensors in autonomous robotics.

In [18], a people detection algorithm for RGB-D data is proposed, which exploits a combination of *Histogram of Oriented Gradients* (HOG) and *Histogram of Oriented Depth* (HOD) descriptors. However, each RGB-D frame is densely scanned to search for people, thus requiring a GPU implementation for being executed in real time. Also [19] and [20] rely on a dense GPU-based object detection, while [21] investigates how the usage of the people detector can be reduced using a depth-based tracking of some *Regions Of Interest* (ROIs). However, the obtained ROIs are again densely scanned by a GPU-based people detector.

In [22], a tracking algorithm on RGB-D data is proposed, which 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.

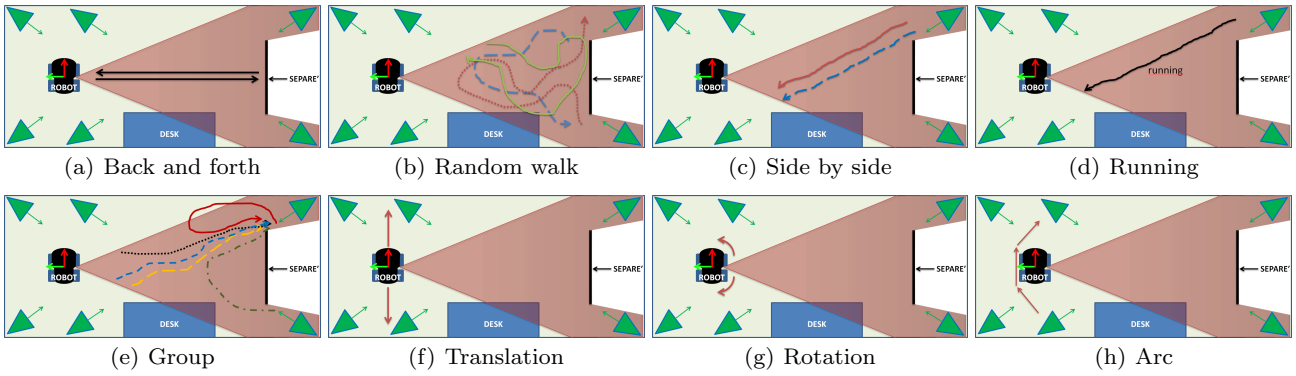
Unfortunately, despite the increasing number of RGB-D people detectors and people tracking algorithms proposed so far, only one data set is available for testing these applications with consumer RGB-D cameras, but this is not suitable for mobile robotics because it only presents images from static Kinect sensors.

1.1 RGB-D datasets

Before Kinect release, the most popular datasets for evaluating people detection and tracking algorithms which exploited aligned RGB and depth data were acquired from stereo cameras or reciprocally calibrated laser range finders and colour cameras. That is the case of [10], which proposed videos acquired from a stereo pair mounted on a mobile platform in order to evaluate people detection in an outdoor scenario, or [23], where a dataset collected with Willow Garage PR2 robot is presented, with the purpose of training and testing multi-modal person detection and tracking in indoor office environments by means of stereo cameras and laser range finders. More recently, [24] and [25] proposed RGB-D datasets acquired with 3D-lidar scanners and cameras mounted on a car.

Since Microsoft Kinect has been introduced, new datasets have been created in order to provide its aligned RGB and depth streams for a variety of indoor applications. [26], [27] and [28] proposed datasets acquired with Kinect suitable for object recognition and pose estimation, while [29] and [30] describe datasets expressly thought for RGB-D action recognition. In [31], the authors propose a new dataset for creating a benchmark of RGB-D SLAM algorithms and in [32] and [33] Kinect data have been released for evaluating scene labeling algorithms. For people tracking evaluation, the only dataset acquired with native RGB-D sensors is proposed in [18] and [22]. The authors recorded data in a university hall from three static Kinects with adjacent, but non overlapping field of view and tracking performance can be evaluated in terms of accuracy (false positives, false negatives, ID switches) and precision in localizing a person inside the image.

However, this dataset is not exhaustive for mobile robotics applications. Firstly, RGB-D data are recorded from a static platform, thus robustness to camera vibrations, motion blur and odometry errors cannot be evaluated, secondly, a 3D ground-truth is not reported, i.e. the actual position of people neither in the robot frame of reference nor in the world frame of reference is known. For many applications, and in particular when dealing with a multi-camera scenario, it becomes also



(a) Back and forth (b) Random walk (c) Side by side (d) Running

(e) Group (f) Translation (g) Rotation (h) Arc

Fig. 2 Illustration of (a-e) the five situations featured in the KTP Dataset and (f-h) the three movements the robot performed inside the motion capture room. Motion capture cameras are drawn as green triangles, while Kinect field of view is represented as a red cone.

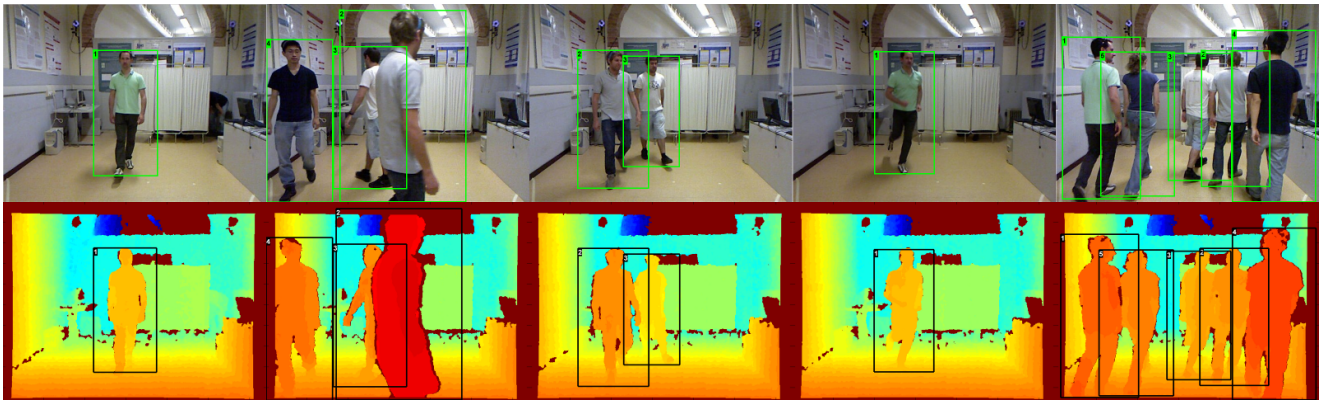


Fig. 3 RGB and Depth images showing the five situations of the *KTP Dataset*, together with the corresponding image annotations.

important to evaluate how accurate and precise a tracking algorithm is in 3D coordinates. In [20], a sort of 3D ground truth is inferred from the image bounding boxes and the depth images computed from stereo data, but these measures are correlated to the sensor depth estimate.

In this work, we describe and extensively evaluate our multi-people tracking algorithm with RGB-D data for mobile platforms originally presented in [34] and improved in [35]. Our people detection approach relies on selecting a set of clusters from the point cloud as people candidates which are then processed by a HOG-based people detector applied to the corresponding image patches. 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. Moreover, we propose a new RGB-D dataset with 2D and 3D ground truth for evaluating accuracy and precision of tracking algorithms also in 3D coordinates. We evaluate our tracking results also on the publicly available *RGB-D People Dataset* and *ETH dataset*, reaching state-of-the-art accuracy and superior speed.

The remainder of the paper is organized as follows: in Section 2 the *Kinect Tracking Precision Dataset* is presented, while in Section 3 an overview of the two main blocks of our tracking algorithm is given. The detection phase is described in Section 3.1, while Section 3.2 details the tracking procedure and in Section 4 we describe the tests performed and we report the results evaluated with the CLEAR MOT metrics [36]. Conclusions and future works are contained in Section 5.

2 The Kinect Tracking Precision Dataset

We propose here a new RGB-D dataset called *Kinect Tracking Precision (KTP) Dataset*[1] acquired from a mobile robot moving in a motion capture room. This dataset has been realized to measure 2D/3D accuracy

[1] `http://www.dei.unipd.it/~munaro/KTP-dataset.html`.

and precision of people tracking algorithms based on data coming from consumer RGB-D sensors.

2.1 Data collection and ground truthing

We collected 8475 frames of a Microsoft Kinect at 640x480 pixel resolution and at 30Hz, for a total of 14766 instances of people. The Kinect was mounted on a mobile robot moving inside a 3x5 meters room equipped with a BTS[2] marker-based motion capture system composed of six infrared cameras. The spatial extent of the dataset was limited by the dimensions of the motion capture room. The dataset provides RGB and depth images, with the depth images already registered to the RGB ones, and robot odometry. They are made available both as single files with timestamp and as ROS `bag` files, that are recordings containing RGB, depth, odometry and transforms among reference frames as a synchronized stream. While Kinect data have been published at 30 frames per second, the robot pose was limited to 10 frames per second. As ground truth, image and 3D people positions are given, together with a further ground truth for robot odometry obtained by placing some markers also on the robot. Image ground truth is in the form of bounding boxes and has been created with the annotating tool and the procedure described in [37]. We annotated only people who are at least half visible in the RGB image. Except when people are partially out of the image, we made the bounding boxes width to be half of the height and centered on the person's head.

3D ground truth consists of people 3D position obtained by placing one infrared marker on every person's head and tracking them with the motion capture system. Then, we referred people 3D position to the robot odometry reference frame and we synchronized the time with that of the images. At this point, we attempted to assign a 3D ground truth to every acquired Kinect frame. When some people were missing (out of the field of view or fully occluded) in the image ground truth, they have been deleted also in the 3D ground truth. When some people were present in the image ground truth, but missing in the 3D ground truth because of occlusions, no 3D ground truth have been associated to those frames. With this process, we assigned 3D ground truth to about 70% of the acquired Kinect frames. In Table 1, some statistics are reported about the image and 3D ground truth.

[2] `http://www.btsbioengineering.com`.

Table 1 Statistics of the ground truth provided with the *KTP Dataset*.

	Image	**3D**
Annotated frames	8475	6287
Frames with people	7058	4870
People instances	14766	10410
Number of tracks	20	20

2.2 Content description

The dataset consists of four videos of about one minute each. In each video, the same five situations are performed:

- *back and forth*: a person walks back and forth once;
- *random*: three persons walk with random trajectories for about 20 seconds;
- *side-by-side*: two persons walk side-by-side with a linear trajectory;
- *running*: one person runs across the room;
- *group*: five persons gather in a group and then leave the room.

The robot moves differently in every video, in order to test tracking performance for different robot motions. The four videos are named according to the movement the robot performs: *Still*, *Translation*, *Rotation*, *Arc*. The robot maximum translation and rotation speeds have been respectively set to 0.15 m/s and 0.11 rad/s for avoiding stability issues due to the high friction produced by the plastic floor of the motion capture room on the robot wheels. In Fig. 2, a pictorial representation of (a-e) the five situations contained in the dataset and (f-h) the movements the robot performed inside the motion capture room is reported. In Fig. 3 some annotated RGB and depth images are reported as representative of the five situations characterizing the dataset.

2.3 Robotic platform

The dataset has been collected with the mobile robot represented in Fig. 1 (b). It consists of a Pioneer P3-AT platform equipped with a side mounted Kinect sensor. With the *KTP Dataset*, we provide both the odometry of this robot and its real position in 3D measured with the motion capture system. This double source of information allowed us to estimate the errors in robot odometry taking the motion capture measurements as ground truth. As we expected, the error in x-y is maximum (22mm) when the robot both translates and rotates for performing an arc movement, while the maximum yaw error (1°) is reached when the robot performs more rotations (namely in *Rotation*). In this work, odometry

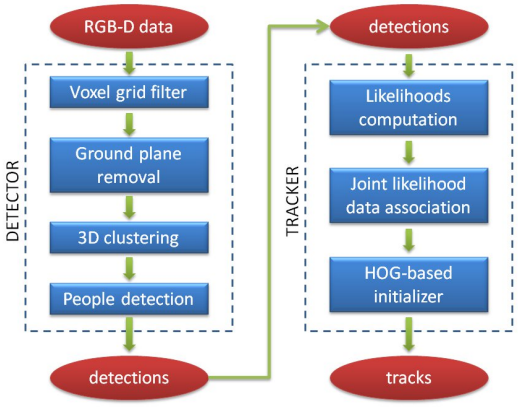


Fig. 4 Block diagram describing input/output data and the main operations performed by our detection and tracking modules.

readings are used to refer people detections to a common (world) reference frame used by the tracking algorithm.

3 People Tracking Algorithm

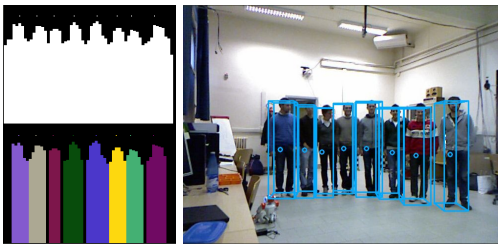
In this work, we thoroughly test the people tracking system we proposed in [35]. As reported in Fig. 4, the RGB-D data are processed by a detection module that filters the point cloud data, removes the ground and performs a 3D clustering of the remaining points. Furthermore, we apply a HOG-based people detection algorithm to the projection onto the RGB image of the 3D clusters extended till the ground, 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.

3.1 Detection

3.1.1 Sub-clustering groups of people

As a pre-processing step of our people detection algorithm, we downsize the input pointcloud by applying a voxel grid filter, which 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. 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.The plane coefficients are computed with a RANSAC-based least square method and they are 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.



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

Fig. 5 Sub-clustering of a cluster containing eight people standing very close to each other.

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, as in [34]. 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 remove clusters too high with respect to the ground plane and merge clusters that are very near in ground plane coordinates, 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 sub-clustering

algorithm presented in [35], that detects the heads from a cluster of 3D points and segment it into sub-clusters according to the head positions. In Fig. 5, 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 black and white height map which contains in every bin the maximum heights from the ground plane of the points of the original cluster, the estimated head positions 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.

3.1.2 HOG detection on clusters extended to the ground

For the sub-clusters obtained, we apply a HOG people detector [1] to the part of the RGB image corresponding to the cluster theoretical bounding box, namely the bounding box with fixed aspect ratio that should contain the whole person, from the head to the ground. It is worth noting that this procedure allows to obtain a more reliable HOG confidence when a person is occluded, with respect to applying the HOG detector directly to the cluster bounding box. As person position, we take the centroid of the cluster points belonging to the head of the person and we add 10cm in the viewpoint direction, in order to take into account that the cluster only contains points of the person surface.

For the people detector, we used Dollár's implementation of HOG[3] and the same procedure and parameters described by Dalal and Triggs [1] for training the detector with the *INRIA Person Dataset* [38]. In Fig. 6, we report the performance of our people detection module evaluated on the *KTP dataset*. The DET curve [37] relates the number of False Positives Per Frame (FPPF) with the False Rejection Rate (FRR) and compares the detection performance with that obtained by applying also the tracking algorithm on top of it. The ideal working point would be in the bottom-left corner (FPPF = 0, FRR = 0%). From the figure, it can be noticed that the tracker performs better than the detector for FPPF > 0.001.

We released an implementation of our people detection algorithm as part of the open source *Point Cloud Library* ([39], [40]), in order to allow comparisons with future works and its use by the robotics and vision communities.

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

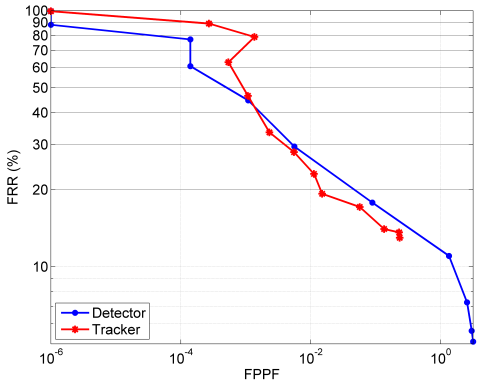


Fig. 6 DET curve comparing the detector and tracker performance on the *KTP Dataset* in terms of False Positives Per Frame (FPPF) and False Rejection Rate (FRR).

3.2 Tracking

The tracking module receives as input 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 motion (in ground plane coordinates) and color appearance, together with that of being a person.

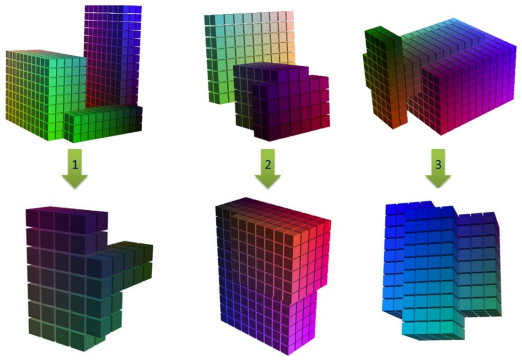
3.2.1 Online classifier for learning color appearance

For every initialized track, we maintain an online classifier based on Adaboost, like the one used in [41] or [22]. 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:

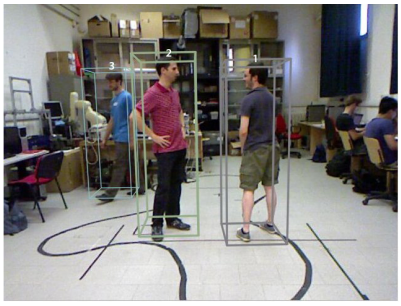
1. we compute the color histogram ($\mathcal{H}$) of the points corresponding to the current detection associated to the track. This histogram can be computed in RGB, HSV or other color space; here, we assume to work on the RGB space. If B is the number of bins chosen, 16 by default, then
$$\mathcal{H} : [1...B]\text{x}[1...B]\text{x}[1...B] \rightarrow \mathbb{N} \quad (1)$$
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. If B_R, B_G and B_B are the bins ranges corresponding to the R, G and B channels, the feature is computed as
$$f(\mathcal{H}, B_R, B_G, B_B) = \sum_{i \in B_R} \sum_{j \in B_G} \sum_{k \in B_B} \mathcal{H}(i,j,k). \quad (2)$$

With this approach, the color histogram is computed only once per frame for all the feature computations. In Fig. 7(a) we report the three most weighted features

(parallelepipeds in the RGB color space) for each one of the three people of Fig. 7(b) 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.



(a)



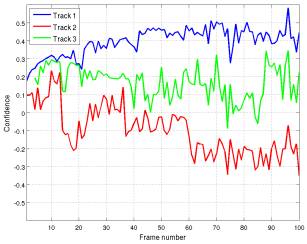
(b)

Fig. 7 (a) 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 shown in (b).

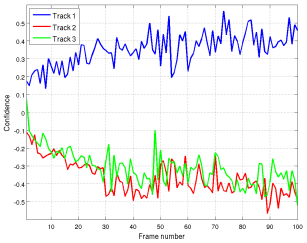
For the training phase, we use as positive sample the color histogram of the target, but, instead of selecting negative examples only from randomly selected windows of the image as in [41], we consider also as negative examples the histograms calculated on the detections not associated to the current track. This approach has the advantage of selecting only the colors that really characterize the target and distinguish it from all the others. Fig. 8 clearly shows how this method increases the distance between the confidences of the correct track and the other tracks.

3.2.2 Three-term joint likelihood

For performing data association, we use the Global Nearest Neighbor approach (solved with the Munkres algorithm), described in [42] and [8]. Our cost matrix derives from the evaluation of a three-term joint likelihood for every target-detection couple.



(a) Random windows.



(b) Other detections and random windows.

Fig. 8 Confidence obtained by applying to the three people of Fig. 7(b) the color classifier trained on one of them (Track 1) for two different methods of choosing the negative examples.

As motion term, we compute the Mahalanobis distance between track i and detection j as

$$D_M^{i,j} = \tilde{\mathbf{z}}_k^{\mathrm{T}}(i,j) \cdot \mathbf{S}_k^{-1}(i) \cdot \tilde{\mathbf{z}}_k(i,j) \tag{3}$$

where $\mathbf{S}_k(i)$ is the covariance matrix of track i provided by a filter and $\tilde{\mathbf{z}}_k(i,j)$ is the residual vector between measurement vector based on detection j and output prediction vector for track i:

$$\tilde{\mathbf{z}}_k(i,j) = \mathbf{z}_k(i,j) - \hat{\mathbf{z}}_{k|k-1}(i). \tag{4}$$

The values we compare with the Mahalanobis distance represent people positions and velocities in ground plane coordinates. Given a track i and a detection j, the measurement vector $\mathbf{z}_k(i,j)$ is composed by the position of detection j and the velocity that track i would have if detection j were associated to it.

An Unscented Kalman Filter is exploited to predict people positions and velocities along the two ground plane axes (x, y). For human motion estimation, this filter turns out to have estimation capabilities near those of a particle filter with much smaller computational burden, comparable to that of an Extended Kalman Filter, as reported by [8]. As people motion model we chose a constant velocity model because it is good at managing full occlusions, as described in [8].

Given that the Mahalanobis distance for multinormal distributions is distributed as a chi-square [43], we use this distribution for defining a gating function for the possible associations.

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

$$L^{i,j}_{TOT} = L^{i,j}_{motion} \cdot L^{i,j}_{color} \cdot L^{j}_{detector}. \quad (5)$$

For simpler algebra we actually minimize the log-likelihood

$$l^{i,j}_{TOT} = -log\left(L^{i,j}_{TOT}\right) = \gamma \cdot D^{i,j}_{M} + \alpha \cdot c^{i,j}_{online} + \beta \cdot c^{j}_{HOG}, \quad (6)$$

where $D^{i,j}_{M}$ is the Mahalanobis distance between track i and detection j, $c^{i,j}_{online}$ is the confidence of the online classifier of track i evaluated with the histogram of detection j, c^{j}_{HOG} is the HOG confidence of detection j and γ, α and β are weighting parameters empirically chosen.

4 Experiments

4.1 Tests with the KTP Dataset

In the following paragraphs, we report a detailed study we perfomed on the RGB-D videos of the *KTP Dataset*. We computed results for every dataset sequence and we studied how different parameters and implementation choices can influence the tracking results.

4.1.1 Evaluation procedure

For the purpose of evaluating the tracking performance we adopted the CLEAR MOT metrics [36], that consists of two indices: MOTP and MOTA. 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, while the MOTP indicator measures how well exact positions of people are estimated. We computed them with formulas reported in [35]. The higher these indices are, the better is the tracking performance. When evaluating the tracking results with respect to the 2D ground truth referred to the image, we computed the MOTP index as the average PASCAL index [44] (intersection over union of bounding boxes) of the associations between ground truth and tracker results by setting the validation threshold to 0.5. Instead, when comparing people 3D position estimates with the ground truth obtained with the motion capture system, we computed the MOTP index as the mean 3D distance for the results that are correctly associated to the ground truth. Since we were interested in evaluating estimates of people position within the ground plane indipendently from people's height estimates, we computed distances of only the x and y coordinates, disregarding the z. We considered an estimate to match with the ground truth if their distance was below 0.3 meters, which seemed to be a fair 3D extension of the area ratio threshold of 0.5 used for the PASCAL test.

From now on, we will use the term *2D* tracking results when referring to tracking results computed on the image, while we will write *3D* tracking results when referring to the evalutation performed in 3D coordinates.

Table 2 Tracking results for the *KTP Dataset* with different algorithms.

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**
Full	84.24%	86.1%	0.8%	12.7%	60
No sub.	84.2%	83.02%	0.6%	15.9%	56
[34]	86%	82.41%	0.9%	16.1%	82

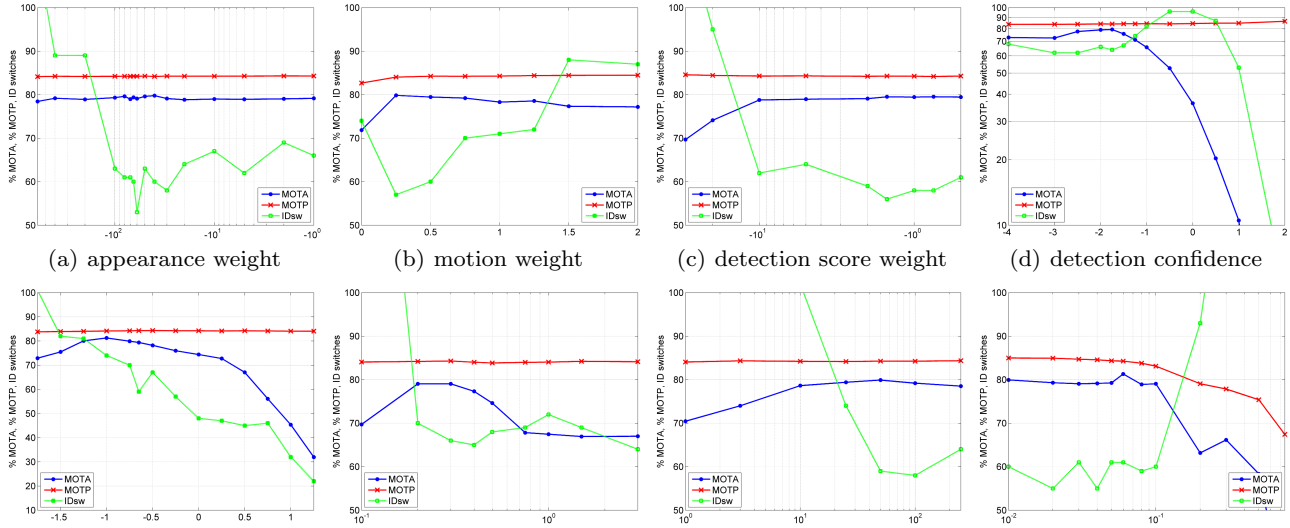
4.1.2 Tracking results

On the first row of Table 2 we report the 2D tracking results we obtained on the KTP dataset with our full algorithm, a variant of our algorithm which does not use the sub-clustering method and the algorithm we presented in [34], which does not perform sub-clustering and uses a different data association procedure. Other than the MOTP and MOTA indices, we report also the false positive and false negatives percentages and the total number of identity (ID) switches, which represents the number of times tracks change ID over time. It can be noticed how the sub-clustering algorithm allowed to separate people very close (mostly present in the *Side-by-side* and *Group* situations), thus reducing the percentage of false negatives. It should be noted that some false positives for the *Group* video are generated by tracks positions not perfectly centered on the person. When not using sub-clustering, less detection were produced because people close to each other or touching were merged into a single detection, thus also less false positives were counted. For the same reasons, also slightly less ID switches were produced. The same difference in false negative rate holds when comparing the work described in this paper with the one in [34]. Moreover, with this work we obtain 14% less ID switches, because we introduced in the log-likelihood computation the confidence given by the appearance classifier that in [34] was used, instead, for a re-identification module decoupled from the motion estimate.

In Table 3, we report the 2D tracking results divided by video. It can be noticed that our algorithm reaches very similar tracking performances for static and moving videos, thus proving to be very good at dealing with robot planar movements.

In Table 4, we present results divided by type of sequence performed. As expected, we can notice that the worst result is obtained for the *Group* situation, where people are often visible only for a small portion. It is

Fig. 9 Tracking output on some frames extracted from the *KTP Dataset*. Different colors are used for different IDs. On top of every bounding box the estimated ID and distance from the camera are reported.



(a) appearance weight (b) motion weight (c) detection score weight (d) detection confidence

(e) min confidence for initialization (f) dist. sub-clustering (g) selectors (h) voxel size

Fig. 10 Performance of our algorithm on the *KTP Dataset* when varying its main parameters. MOTA (blue) and MOTP (red) curves represent percentages, while ID switches (green) is the number obtained by summing up the ID switches for every video of the dataset. The optimum is reached when MOTA and MOTP are maximum and ID switches are minimum. In some of them, the logarithmic scale is used for the x or y axis for better visualization. Please, see text for details.

Table 3 Tracking results for the *KTP Dataset* divided by video.

	MOTP	MOTA	FP	FN	ID Sw.
Still	83.76%	88.86%	0.9%	9.8%	15
Translation	84.39%	88.03%	0.8%	10.7%	15
Rotation	84.19%	83.16%	0.9%	15.4%	19
Arc	84.89%	83.24%	0.8%	15.5%	13

Table 4 Tracking results for the *KTP Dataset* divided by situation.

	MOTP	MOTA	FP	FN	ID Sw.
Back and forth	84.23%	89.17%	0.9%	9.9%	1(0)
Random walk	84.4%	86.42%	1%	12.3%	22(0)
Side by side	84.6%	79.12%	0.6%	20.2%	5(5)
Running	80.79%	90.44%	0.9%	8.7%	4(4)
Group	83.59%	63%	0.8%	35.8%	32(16)

worth distinguishing between ID switches caused by an ID previously associated to a track and then to another and those generated when a person exits the scene and re-appears after several seconds. In this table, we reported the total number of ID switches and then we wrote inside the parenthesis the number of ID switches due to the latter reason. We can notice that the random walk sequences produce the highest number of ID switches of the first type. This fact can be explained because we use a constant velocity model to predict people position. This model results to work well for all the other situations, but performs worse in the *Random walk* situation because people abruptly change their direction. The constant velocity model is also not suited for avoiding ID switches of the second type, because it leads to a very low likelihood for people who exit the room going in one direction and re-enter the room walking towards the opposite direction. In Table 5, we report the same table, but using the 3D evaluation method. We can notice that MOTA changes considerably, increasing for the *Syde-by-side* and *Running* situation, but decreasing for the *Random walk* and *Group* ones. The *Group* situation, other than for very strong occlusions, is also challenging because three people enter the field of view of the camera from close to the robot and are not fully visible in the image for some seconds.

Table 5 3D tracking results for the *KTP Dataset* divided by situation.

	MOTP	MOTA	FP	FN	ID Sw.
Back and forth	0.196m	88.97%	2.4%	8.5%	1(0)
Random walk	0.171m	70.93%	9.8%	18.9%	20(0)
Side by side	0.146m	87.22%	1.2%	11.6%	5(5)
Running	0.143m	94.57%	1.1%	4.4%	4(4)
Group	0.181m	47.91%	9.1%	42.53%	26(16)

Table 6 Tracking results for different colorspaces.

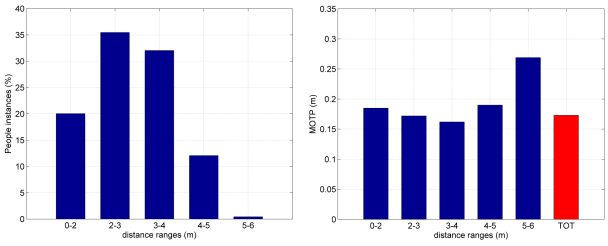
	MOTP	MOTA	FP	FN	ID Sw.
RGB	84.24%	86.1%	0.8%	12.7%	60
HSV	84.22%	85.8%	0.7%	12.5%	53
CIELab	84.22%	86.48%	0.9%	12.2%	56
CIELuv	84.25%	86.72%	0.9%	12.9%	65

In Fig. 9, we report also some examples of tracked frames relative to the KTP dataset. 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 partially occluded.

We evaluated different colorspaces to be used for computing the color histogram of the clusters. As it can be seen in Table 6, the HSV space turned out to work better than RGB, CIELab and CIELuv, especially for reducing the number of ID switches.

The Microsoft Kinect estimates distances by means of a triangulation method. That means that its precision decreases with the squared distance from the camera. In order to analyze accuracy and precision of our tracking algorithm at various distances, we report in Fig. 11(a) the percentage of people instances and in Fig. 11(b) the 3D tracking results obtained on our dataset for different distance ranges from the camera. If a person is too close to the camera, the head could be not visible, thus the tracking algorithm could produce a worse position estimate. For this reason, the optimal range of distances for people tracking with Kinect turned out to be of 3-4 meters. Under five meters, the mean tracking precision is below 20 centimeters, which is a fair localization error for robotics applications, while, over five meters, the MOTP rapidly increases, because Kinect depth estimates lose in accuracy.

In Fig. 10, we report graphs of MOTA, MOTP and ID switches deriving from the 2D evaluation of our algorithm while varying its main parameters. The optimum is reached when MOTA and MOTP are maximum and ID switches are minimum. The (a-c) graphs show the effects of varying the weights given respectively to the color, motion and detector likelihoods. It can be noticed how MOTP is almost invariant to these parameters, thus the main criteria for choosing the weights is to look at MOTA and ID switches. In (d), we report results



(a) (b)

Fig. 11 (a) Percentage of people instances for variuos distance ranges from the sensor and (b) MOTP value for every distance range.

for various values of the minimum confidence for people detection. Minimum confidences under -1.25 work well, while above this value the performance rapidly worsen because many people are missed. For what concerns the track initialization, a value between -1 and -0.5 seems to be the best choice. Below this value some false tracks are generated, above this value many people instances are missed. The results reported in (f) confirm that the intimate distance (0.3 meters) is the best choice for the minimum distance between people to be used in the sub-clustering method. If this value is too low, false positives are generated because a person can result splitted in many clusters. Instead, if using a too high threshold, people close to each other would remain merged in a single cluster.

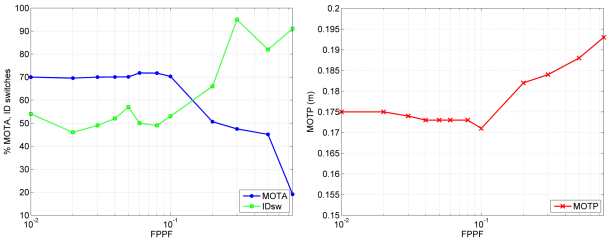
In order to test our appearance classifier, we varied the number of features (selectors) that are chosen at every iteration from a pool of 250 weak classifiers for composing Adaboost strong classifier. It can be seen in (g) that selecting from 50 to 100 features from a pool of 250 gives the best tracking results. Given that the higher this number, the higher the computational cost, 50 has been chosen as the default parameter. As a further test for evaluating the effectiveness of using an online learning scheme for computing the appearance likelihood needed for data association, we compared our results with a method which does not exploit learning and keeps the person classifier fixed after its initialization. With this approach, we measured a MOTA decrease of 7%, in particular due to an increased number of ID switches and missed people.

Finally, an important trade-off between precision and computational speed must be taken into account for the choice of the voxel size. The higher this value is, the faster is the algorithm because less points have to be processed, but precision is lost in estimating people position. From the graph in (h) it can be noticed that the algorithm still performs well with a voxel size of 0.1 meters, then the performance degrades very quickly, this time involving also the MOTP index. Moreover, we

Table 7 Best parameters values resulting from our study.

Parameter	Value
Appearance weight	[30; 100]
Motion weight	0.5
Detection weight	1
Detection confidence	< -1.25
Min confidence for initialization	[-1; -0.5]
Distance for sub-clustering	0.3
Number of selectors	50
Voxel size	< 0.1
Colorspace	HSV

show in Fig. 12 how the voxel size impacts on the 3D tracking results. Even when evaluating the results in 3D coordinates, we could see that until a voxel size of 0.1 meters the obtained results are good, while, for bigger values, all the three indices rapidly get worse.



(a) (b)

Fig. 12 3D evalutation on the *KTP Dataset* when varying the voxel size.

In Table 7, we summarize the parameters values of our detection and tracking algorithm which gave the best tracking results on the *KTP Dataset*.

4.2 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* ([45], [18], [22]), that contains about 4500 RGB-D frames acquired from three vertically mounted Kinect sensors. For this test, we used three independent people detection modules (one for each Kinect), then detections have been fused at the tracking stage.

In Table 8, we report the results obtained with our default system against those obtained in [22], which uses GPU computation. A video with our tracking results can be found at this link: `http://youtu.be/b70vLKFsqiM`. Our MOTA and ID switches indices are 71.8% and 19, while for [22] they are 78% and 32, respectively.

For a correct interpretation of these results, the following considerations must be taken into account:

Table 8 Tracking evaluation with RGB-D People Dataset.

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

- 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%;
- 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 noting that half of our false negatives refer to those people, thus reducing to 10% the percentage of missed detections on the ground plane;
- 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. 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%. If we do not consider people on the stairs, the MOTA value raises to 88.9%.

Part of the success of our tracking is due to sub-clustering. In fact, if we do not use the sub-clustering method described in Section 3.1.1, the MOTA index decreases by 10%, while the ID switches increase by 17.

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

4.3 Test with the ETH Dataset

In addition to the experiments we reported with Kinect-style RGB-D sensors, we also tested our algorithm on a part of the challenging *ETH dataset* [10], where data were acquired from a stereo pair mounted on a mobile platform moving at about 1.4 m/s in a densely populated outdoor environment.

In Fig. 13, we show our tracking results on the *Bahnhof* sequence, which is composed of 1000 frames acquired at 14Hz at a resolution of 640x480 pixels. We compare our FPPF vs miss-rate DET curve with those of the state-of-the-art approaches already reported in [20]. We remind that the ideal working point would be in the bottom-left corner (FPPF = 0, FRR = 0%). It

is worth noting that depth data obtained from stereo are more noisy and present more artifacts with respect to those obtainable with Kinect-style sensors. Standard state-of-the-art algorithms highly rely on a dense scanning of the RGB image, thus being less sensible to bad depth data, while our approach processes only patches of the RGB image corresponding to valid depth clusters, thus being more dependent on the quality of depth estimates. Nevertheless, we obtain performance near state-of-the-art, doing better than [10] for FPPF less than 0.1. The algorithm which performs best is a variant of the method in [20] which uses the *Deformable Parts Model* [46] detector. Similar results are also obtained by the standard approaches in [20], [47] and [48]. However, the latter two algorithms performs a global optimization of all the tracked frames, thus requiring all images in a batch as an input. Among all these methods, our approach is the only one which works in real time on a standard laptop CPU. In Fig. 13 (b), we also reported the robot path given by the odometry (in black) and all people trajectories estimated by our algorithm.

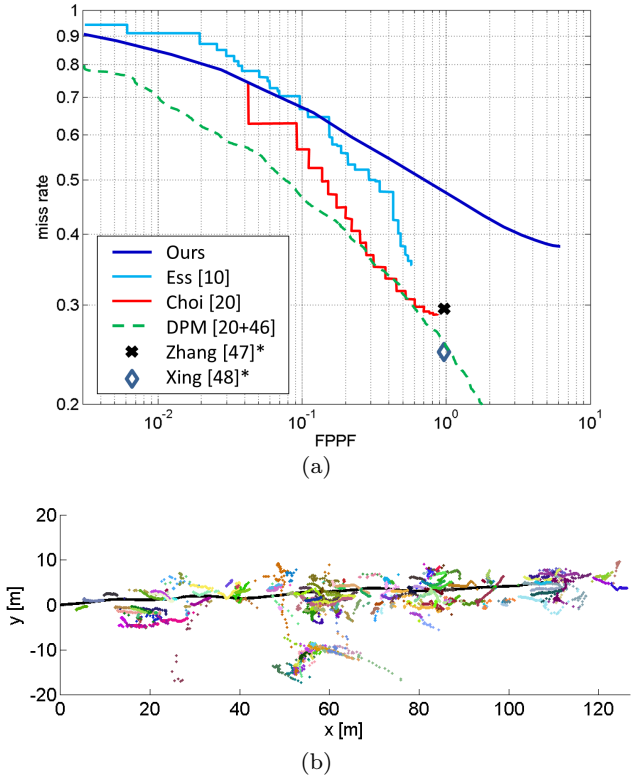


Fig. 13 (a) FPPF vs miss-rate curve showing tracking results on the *Bahnhof* sequence of the *ETH dataset*. The papers with * require all the images in a batch as an input. (b) all estimated people trajectories (various colors) and the robot trajectory (in black).

4.4 People following tests

For proving the robustness and the real time capabilities of our tracking method, we tested the system on board of a service robot moving in crowded indoor environments. The robot's task was to follow a specific tracked person and its speed was only limited by the manufacturer to 0.7 m/s. We tested our whole system both in a laboratory setting and in a crowded environment at the Italian robotics fair *Robotica 2011*, held in Milan on the 16-19th November 2011. 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. In Fig. 14, we report some tracking results while our robot was following a person along a narrow corridor with many light changes (first row) or when other people were walking near the person to follow (second row). Finally, also some tracking results collected at *Robotica 2011* fair are shown (third row).

4.5 Runtime perfomance

Our system has been developed in `C++` and `Python` (for the people following module) with ROS, the Robot Operating System [49], making use of highly optimized libraries for 2D computer vision [50], 3D point cloud processing [39] and bayesian estimation[4].

In Table 9, 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[5]. These frame rates are achieved by using Kinect QQVGA depth resolution[6], while they halve at Kinect VGA resolution. The most demanding operations are Euclidean clustering (Section 3) and HOG descriptors computation (Section 3.1.2), which require 46% and 23% of time with QQVGA resolution, respectively. The tracking algorithm is less onerous since it occupies 8-17% of the CPU.

Our implementation does not rely on GPU processing, nevertheless, our overall algorithm is faster than other state-of-the-art algorithms such as [11], [21] and [20], respectively running at 3, 10 and 10 fps with GPU processing. This suggests that even a robot with limited computational resources could use the same computer for people tracking and other tasks like navigation and

[4] Bayes++ - `http://bayesclasses.sourceforge.net`.
[5] Both computers had 4GB DDR3 memory.
[6] This is the resolution used for most of the tests reported in this paper.

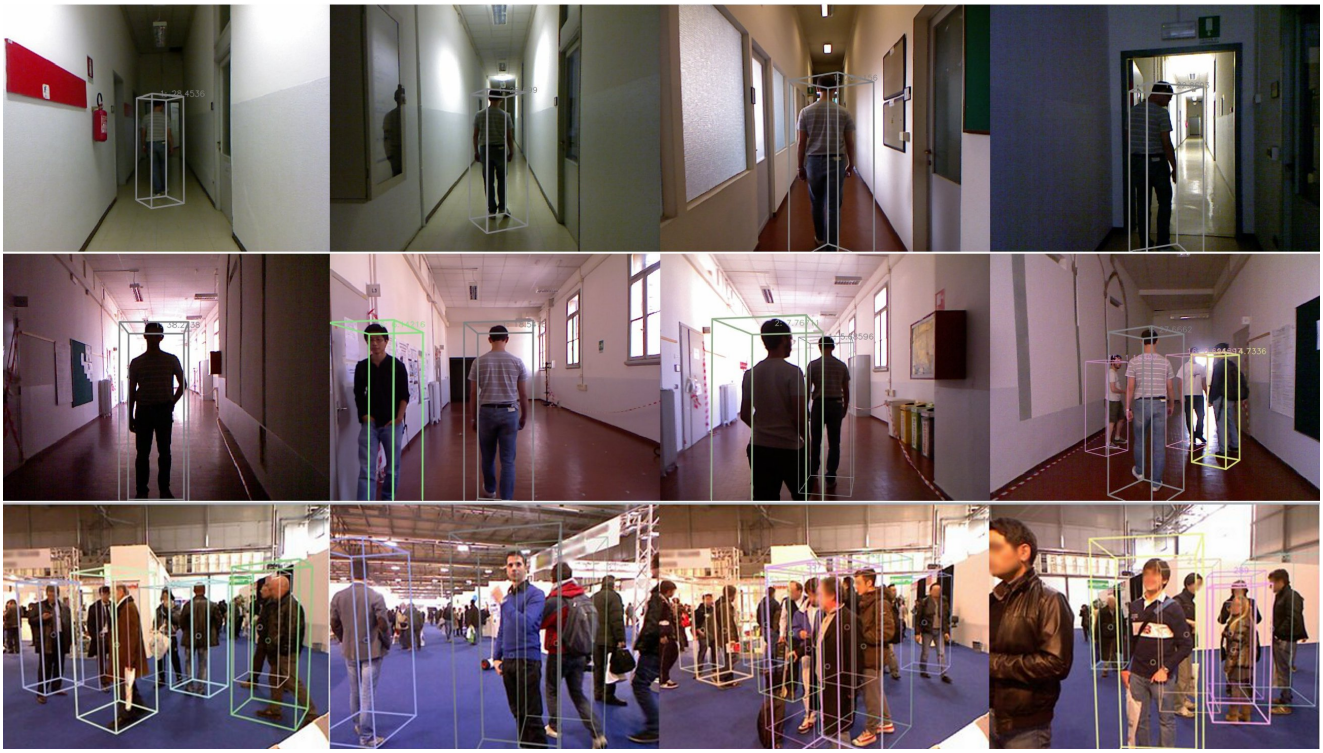


Fig. 14 People following tests. First row: examples of tracked frames while a person is robustly followed along a narrow corridor with many light changes. Second row: other examples of correctly tracked frames when other people are present in the scene. Third row: tracking results from our mobile robot moving in a crowded environment. In these tests, the top speed of the robot was 0.7 m/s.

Table 9 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

self localization. It is worth noting that also our algorithm would benefit from GPU computing. For example, voxel grid filtering, Euclidean clustering and HOG descriptors computation, which take about 80% of the computation, are all highly parallelizable.

We exploit ROS multi-threading capabilities and we explicitly designed our system for real time operation and for correctly handling data coming from multiple sensors, delays and lost frames. Please refer to [35] and [51] for a description of the tests we performed with multiple Kinects in a centralized and distributed fashion.

5 Conclusions and future works

In this paper, we presented a fast and robust algorithm for multi-people tracking with RGB-D data designed to be used on mobile service robots. It can track multiple people with state-of-the-art accuracy and beyond state-of-the-art speed without relying on GPU computation. Morever, we introduced the *Kinect Tracking Precision Dataset*, the first RGB-D dataset with 2D and 3D ground truth for evalutating accuracy and precision of people tracking algorithms also in terms of 3D coordinates and we found that the average 3D error of our people tracking system is lower enough for robotics applications. From the extensive evaluation of our method on this dataset and on another public dataset acquired from three static Kinects, we demonstrated that Kinect-style sensors can replace sensors previously used for indoor people tracking from service robotics platforms, such as stereo cameras and laser range finders, while reducing the required computational burden.

As a future work, we plan to improve the motion model we use for predicting people trajectories, by combining random walk and social force models, and to test how much gain in accuracy and loss in speed would cause an extension to multiple hypothesis of our tracking algorithm. Moreover, we aim to realize novel people identification methods based on RGB-D cues and use them in our tracking scheme.

Acknowledgment

We wish to thank the Biongineering of Movement Laboratory of the University of Padova for providing the motion capture facility, in particular Martina Negretto and Annamaria Guiotto for their help for the data acquisition and all the people who took part to the *KTP Dataset*. We wish also to thank Filippo Basso and Stefano Michieletto as co-authors of the previous publications related to this work and Mauro Antonello for the advices on the disparity computation for the *ETH dataset*.

References

1. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR 2005*, vol. 1, June 2005, pp. 886–893.
2. 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 *ICCV 2009*, vol. 1, October 2009, pp. 1515–1522.
3. O. Mozos, R. Kurazume, and T. Hasegawa, "Multi-part people detection using 2d range data," *International Journal of Social Robotics*, vol. 2, pp. 31–40, 2010.
4. L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart, "A layered approach to people detection in 3d range data," in *AAAI'10, PGAI Track*, Atlanta, USA, 2010.
5. L. Spinello, M. Luber, and K. O. Arras, "Tracking people in 3d using a bottom-up top-down people detector," in *ICRA 2011*, Shanghai, 2011, pp. 1304–1310.
6. A. Carballo, A. Ohya, and S. Yuta, "Reliable people detection using range and intensity data from multiple layers of laser range finders on a mobile robot," *International Journal of Social Robotics*, vol. 3, no. 2, pp. 167–186, 2011.
7. L. E. Navarro-Serment, C. Mertz, and M. Hebert, "Pedestrian detection and tracking using three-dimensional ladar data," in *FSR*, 2009, pp. 103–112.
8. N. Bellotto and H. Hu, "Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters," *Autonomous Robots*, vol. 28, pp. 425–438, May 2010.
9. 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*, vol. 54, no. 9, pp. 721–728, 2006.
10. A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *CVPR 2008*, 2008, pp. 1–8.
11. ——, "Moving obstacle detection in highly dynamic scenes," in *ICRA 2009*, 2009, pp. 4451–4458.
12. 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*, vol. 28, no. 11-12, 2009, pp. 1466–1485.
13. J. Satake and J. Miura, "Robust stereo-based person detection and tracking for a person following robot," in *Workshop on People Detection and Tracking (ICRA 2009)*, 2009.
14. http://www.microsoft.com/en-us/kinectforwindows.
15. http://dinast.com/cyclopes-od.
16. http://www.pmdtec.com/products_services/pmd_photonics_specs.php.
17. 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 *ISSCC 2012*, San Francisco, USA, February 2012, pp. 392–394.
18. L. Spinello and K. O. Arras, "People detection in rgb-d data," in *IROS 2011*, 2011, pp. 3838–3843.
19. W. Choi, C. Pantofaru, and S. Savarese, "Detecting and tracking people using an rgb-d camera via multiple detector fusion," in *ICCV Workshops 2011*, 2011, pp. 1076–1083.
20. ——, "A general framework for tracking multiple people from a moving camera," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 7, pp. 1577–1591, 2012.
21. D. Mitzel and B. Leibe, "Real-time multi-person tracking with detector assisted structure propagation," in *ICCV Workshops 2011*, 2011, pp. 974–981.
22. M. Luber, L. Spinello, and K. O. Arras, "People tracking in rgb-d data with on-line boosted target models," in *IROS 2011*, 2011, pp. 3844–3849.
23. C. Pantofaru, "The Moving People, Moving Platform Dataset," http://bags.willowgarage.com/downloads/people_dataset/.
24. G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, November 2011.
25. A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR 2012*, Providence, USA, June 2012, pp. 3354–3361.
26. K. Lai, L. Bo, X. Ren, , and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *ICRA 2011*, May 2011, pp. 1817–1824.
27. A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell, "A Category-Level 3-D Object Dataset: Putting the Kinect to Work," in *ICCV Workshop on Consumer Depth Cameras in Computer Vision*, November 2011.
28. http://solutionsinperception.org/.
29. J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured Human Activity Detection from RGBD Images," in *ICRA 2012*, May 2012, pp. 842–849.
30. H. Zhang and L. E. Parker, "4-dimensional local spatio-temporal features for human activity recognition," in *IROS 2011*, sept. 2011, pp. 2044–2049.
31. J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IROS 2012*, Oct. 2012, pp. 573–580.
32. H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *NIPS*, 2011, pp. 244–252.
33. N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *ICCV 2011 - Workshop on 3D Representation and Recognition*, 2011, pp. 601–608.
34. F. Basso, M. Munaro, S. Michieletto, E. Pagello, , and E. Menegatti, "Fast and robust multi-people tracking from rgb-d data for a mobile robot," in *IAS-12*, Jeju Island, Korea, June 2012, pp. 265–276.
35. M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with rgb-d data," in *IROS 2012*, Algarve, Portugal, October 2012, pp. 2101–2107.

36. K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal of Image Video Processing*, vol. 2008, pp. 1:1–1:10, January 2008.
37. P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR 2009*, 2009, pp. 304–311.
38. http://pascal.inrialpes.fr/data/human.
39. R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *ICRA 2011*, Shanghai, China, May 9-13 2011, pp. 1–4.
40. http://pointclouds.org/documentation/tutorials/ground_based_rgbd_people_detection.php.
41. H. Grabner and H. Bischof, "On-line boosting and vision," in *CVPR*, vol. 1. IEEE Computer Society, 2006, pp. 260–267.
42. P. Konstantinova, A. Udvarev, and T. Semerdjiev, "A study of a target tracking algorithm using global nearest neighbor approach," in *CompSysTec 2003: e-Learning*. ACM, 2003, pp. 290–295.
43. http://www.ime.unicamp.br/~cnaber/mvnprop.pdf.
44. M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.
45. http://www.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html.
46. P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 9, pp. 1627–1645, September 2010.
47. L. Zhang, Y. Li, and N. R., "Global data association for multi-object tracking using network flows," in *CVPR*, 2008, pp. 1–8.
48. J. Xing, H. Ai, and S. Lao, "Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses." in *CVPR*, 2009, pp. 1200–1207.
49. M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *ICRA*, 2009.
50. G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
51. M. Munaro, F. Basso, S. Michieletto, E. Pagello, and E. Menegatti, "A software architecture for rgb-d people tracking based on ros framework for a mobile robot," in *Frontiers of Intelligent Autonomous Systems*. Springer, 2013, vol. 466, pp. 53–68.