2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Multisensor Online Transfer Learning for 3D LiDAR-based Human Detection with a Mobile Robot

Zhi Yan[1], Li Sun[2], Tom Duckett[2], and Nicola Bellotto[2]

***Abstract*— Human detection and tracking is an essential task for service robots, where the combined use of multiple sensors has potential advantages that are yet to be fully exploited. In this paper, we introduce a framework allowing a robot to learn a new 3D LiDAR-based human classifier from other sensors over time, taking advantage of a multisensor tracking system. The main innovation is the use of different detectors for existing sensors (i.e. RGB-D camera, 2D LiDAR) to train, online, a new 3D LiDAR-based human classifier based on a new "trajectory probability". Our framework uses this probability to check whether new detection belongs to a human trajectory, estimated by different sensors and/or detectors, and to learn a human classifier in a semi-supervised fashion. The framework has been implemented and tested on a real-world dataset collected by a mobile robot. We present experiments illustrating that our system is able to effectively learn from different sensors and from the environment, and that the performance of the 3D LiDAR-based human classification improves with the number of sensors/detectors used.**

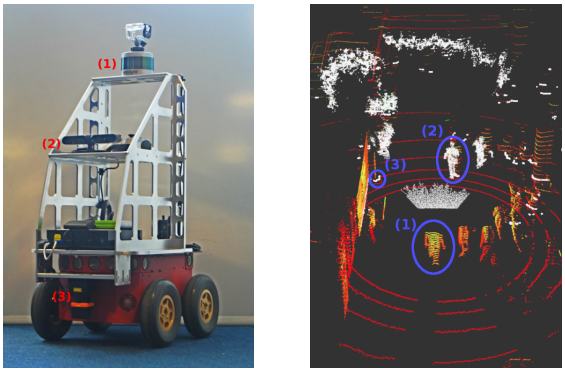

Fig. 1. Left: Pioneer 3-AT mobile robot with Velodyne VLP-16 3D LiDAR (1), ASUS Xtion PRO LIVE RGB-D camera (2), and HOKUYO UTM-30LX 2D LiDAR (3). Right: visualized sensor data (only depth information is shown for RGB-D camera) annotated with corresponding sensor numbers.

I. INTRODUCTION

Human detection and tracking is an important task in service robotics, where knowledge of human motion properties such as position, velocity and direction can be used to improve the behavior of the robot, for example to improve its collision avoidance and adapt its velocity to that of the surrounding people. Using multiple sensors to track people has advantages over a single one. The most obvious one is that multiple sensors can often do the task with a wider field of view and thus track more people within a larger range [1], [2]. Another advantage is that multiple sensors providing redundant information can increase tracking accuracy and reliability [3], [4], [5], [6].

Different sensors have different properties. The 3D LiDAR in our robot platform (Fig. 1) has 16 scan channels, 360° horizontal and 30° vertical fields of view, and up to 100 m range. However, this sensor provides only sparse point clouds, from which human detection can be very difficult because some useful features, such as color and texture, are missing. 2D LiDARs have obviously similar problems, with further limitations due the availability of a single scan channel and reduced field of view. However, these sensors are also cheaper than the previous, and have been used in mobile robotics long enough to stimulate the creation of many human detection algorithms [7], [8]. RGB-D cameras, instead, can detect humans more reliably but only within short range and limited field of view [9], [10].

[1]LE2I-CNRS, University of Technology of Belfort-Montbéliard, France zhi.yan@utbm.fr
[2]Lincoln Centre for Autonomous Systems, University of Lincoln, UK {lsun, tduckett, nbellotto}@lincoln.ac.uk

In the literature, several algorithms already exist which can reliably detect human under particular conditions and with specific sensors (e.g. close range RGB-D detection). Other sensors, however, are not yet so popular to benefit from good human detection software (e.g. 3D LiDAR). In some cases, there are simply not enough datasets with such sensors to learn robust human classifiers for many real-world applications. In this paper, therefore, we wish to train a 3D LiDAR-based human classifier in a semi-supervised way by learning from existing RGB-D and 2D LiDAR-based detectors. Although better human detectors ultimately lead to better people tracking systems, here we focus on the first part only and leave the second for future work.

Typically, data collection and training of the classifier are done offline, with obvious labor cost and potential human errors. In the proposed transfer learning framework, instead, a 3D LiDAR-based human classifier is trained online while exploiting spatio-temporal information from the tracking sub-system, which uses static (i.e. pre-trained) human detectors for the 2D LiDAR and RGB-D sensors. This framework enables a new sensor to learn from trajectories of the tracked people, each one with an associated confidence, or probability, of being human-generated, and fusing different model-based (labeled) and model-free (unlabeled) detections according to a semi-supervised learning scheme [11]. In contrast to previous approaches [12], [13], our solution does not need any hand-labeled data, performing online learning completely from scratch. Besides reducing the burden of data annotation, this feature makes our system easily adaptable to the environment where the robot is deployed.

The contributions of this paper can be summarized as

follows: 1) we propose an online transfer learning framework for multisensor people tracking based on a new *trajectory probability*, which takes into account both sensor independence (in the detection) and multisensor interaction (in the trajectory estimation); 2) we present an experimental evaluation of our system for 3D LiDAR-based human classification with a mobile robot on a real-world dataset using different sensor combinations.

The remainder of the paper is organized as follows: Section II provides an overview of relevant literature in human detection and tracking. Section III presents our solution framework for online transfer learning. Section IV describes the application of the proposed framework to the problem of 3D LiDAR-based human classification. Section V illustrates the experimental results for different sensor configurations. Finally, Section VI concludes the paper summarizing the contributions and suggesting future research work.

II. RELATED WORK

The problem of multitarget and multisensor tracking has been extensively studied during the past few decades. Most of present systems are based on Bayesian methods [14], which compute an estimate of the correspondence between features detected in the sensor data and the different humans to be tracked. Regarding robotic applications, multiple sensors can be deployed in single- or multi-robot systems [15], while the former is the concern of this paper.

RGB/RGB-D camera plus 2D LiDAR is the most frequently used combination in the literature. [3] presented two different methods for mobile robot tracking and following of a fast-moving person in an outdoor environment. The robot was equipped with an omnidirectional camera and a 2D LiDAR. [16] presented an integrated system to detect and track people and cars in outdoor scenarios, based on the information retrieved from a camera and a 2D LiDAR on an autonomous car. [2] introduced a people tracking system for mobile robots in very crowded and dynamic environments. Their system was evaluated with a robot equipped with two RGB-D cameras, a stereo camera and two 2D LiDARs.

Our previous work with the aforementioned sensor combination includes [8] and [17]. The former presented a human tracking system for mobile service robots in populated environments, while the latter extended this system to a fully integrated perception pipeline for people detection and tracking in close vicinity to the robot. The proposed tracking system tracks people by detecting legs extracted from a 2D LiDAR and fusing this with the faces or the upper-bodies detected with a camera using a sequential implementation of the Unscented Kalman Filter (UKF).

The combination with 3D LiDAR is increasing with the development of the 3D LiDAR technology. Taking advantage of its high accuracy, [4] developed an algorithm to align 3D LiDAR data with high-resolution camera images obtained from five cameras, in order to accurately track moving vehicles. Other reported results include [18] and [19], which mainly focused on pedestrian detection rather than tracking. In addition, earlier work presented multitarget tracking with a mobile robot equipped with two 2D LiDARs, respectively located at the front and back [1]. Thus the robot can have a 360° horizontal field of view, where each scan of these two sensors covers the whole surrounding of the robot at an angular resolution of 1°.

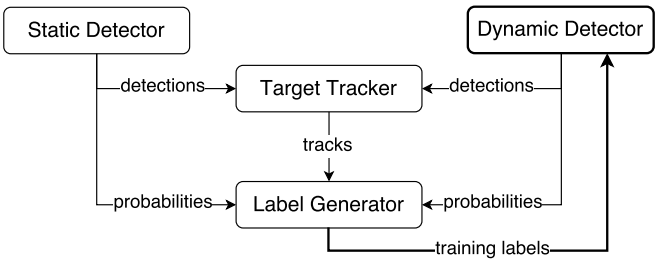

Fig. 2. Block diagram of the online transfer learning framework.

The use of machine learning algorithms for tracking has particular advantages. The closest work to ours is [13], where the authors proposed a semi-supervised learning approach to the problem of track classification in 3D LiDAR data, based on Expectation-Maximization (EM) algorithm. In contrast to our approach, their learning procedure needs a small set of seed tracks and a large set of background tracks, that need to be manually or semi-manually labeled at first, whereas we do not need any hand-labeled data.

To our knowledge, no existing work in the robotics field explicitly exploits information from multisensor-based tracking to implement transfer learning between different sensors as in this paper. Our work combines the advantages of multiple sensors with the efficiency of semi-supervised learning, and integrates them into an single online framework applied to 3D LiDAR-based human detection.

III. ONLINE TRANSFER LEARNING

An overview of our solution framework for online transfer learning can be seen in Fig. 2. It contains four main components: static detectors denoted by $\mathcal{D}_s$, dynamic detectors $\mathcal{D}_d$, a target tracker $\mathcal{T}$ and a label generator $\mathcal{G}$. In order to facilitate the explanation, we present each component following the sequence of an entire iteration, starting with human detection.

A. Human Detection and Tracking

$\mathcal{D}_s$ can detect humans with offline-trained or heuristic detectors, typically with high confidence, while $\mathcal{D}_d$ acquires this ability through the online framework. Both detectors provide new observations for $\mathcal{T}$ and their corresponding probabilities for $\mathcal{G}$. $\mathcal{D}_s$ provides labeled detections, while $\mathcal{D}_d$ provides both labeled and unlabeled ones. Here we assume the initial training set is substituted, instead, by a transfer learning process between the initial $\mathcal{D}_s$ and the final $\mathcal{D}_d$.

The tracking process $\mathcal{T}$ gathers the observations, fuses them and generates human motion estimates. Both moving and stationary targets are tracked. For the latter, the trajectory length is supposed to be null or at least very small. $\mathcal{T}$ associates human detections from different sensors to the same corresponding estimates, linking $\mathcal{D}_s$ and $\mathcal{D}_d$ detections

and therefore making the transfer learning possible. In order to enable this on a mobile robot with multiple sensors, $\mathcal{T}$ should: a) be robust to sensor noise and partial occlusions; b) fuse multisensor data; c) be able to deal with multiple targets simultaneously; d) cope with noise introduced by robot motion.

B. Transfer Learning

The label generator $\mathcal{G}$ fuses the information coming from $\mathcal{D}_s$, $\mathcal{D}_d$ and $\mathcal{T}$, then generates training labels for $\mathcal{D}_d$. A *trajectory probability* is measured by $\mathcal{G}$, based on Bayes' theorem. The idea is to measure the likelihood that a trajectory belongs to a human, which is defined as follow. Given an objectness proposal x_i and its category label y_i, $P(y_i|x_i, d_j)$ denote the predictive probability that sample x_i is a human observed by detector $d_j \in \mathcal{D}$ ($\mathcal{D} = \mathcal{D}_s \cup \mathcal{D}_d$) at time t. For a whole trajectory of detections $X_T\{x_i\}$ and its category label Y_T, the predictive probability of the whole trajectory $P(Y_T|X_T, \mathcal{D})$ is computed by integrating the observations of the different detectors according to the following formula:

$$P(Y_T|X_T, \mathcal{D}) = \frac{odds_{X_T}}{1 + odds_{X_T}} \tag{1}$$

where

$$odds_{X_T} = \prod_{i=1}^{t} \prod_{j=1}^{K} odds_{x_i}^{j} \tag{2}$$

and

$$odds_{x_i}^{j} = \frac{P(y_i|x_i, d_j)}{1 - P(y_i|x_i, d_j)} \tag{3}$$

Here, $P(y_i|x_i, d_j)$ may be necessary to get through some transformations based on specific circumstances, that are illustrated in our application case (Section IV-A and IV-B). The above formula has been proved effective by theoretical and experimental analysis in generating grid occupancy map from different sensors [20], [21], since the odds method takes sensor interactions into account. To the best of our knowledge, applying it to human trajectory analysis is new.

An intuitive approach to generate labels is to threshold the trajectory probability in Eq. 1. Let σ_t be a predefined threshold value, then:

$$\begin{aligned} &\forall x_i \in X_T \text{ assign } y_i = z(z \in \mathbb{Z}^+), \text{ get } X_T, Y_T = \{x_i, y_i\}_T \\ &\qquad \text{if } P(Y_T|X_T, \mathcal{D}) \geq \sigma_t \end{aligned} \tag{4}$$

where $\mathbb{Z}^+$ refers to positive integer values. Our framework uses a Batch Incremental Training (BIT) policy to learn a classifier in $\mathcal{D}_d$, because: a) the number of new generated training examples is always more than one; b) the batch of examples can be used to approximate the distribution of the whole examples in online learning [22].

For a certain interval, a new batch of training samples $[X_{new}, Y_{new}]$ can be generated:

$$\begin{aligned} [X_{new}, Y_{new}] = \{(x_i, y_i) \mid x_i \in X_T \subseteq \mathbb{R}^n, \\ y_i \in Y_T \subseteq \mathbb{Z}^+, 1 \leq i \leq n_i\}_{\forall T} \end{aligned} \tag{5}$$

where X_T is obtained by the tracker and Y_T is obtained from Eq. 4. The classifier of $\mathcal{D}_d$ will be updated in the BIT procedure:

$$\mathcal{D}_d^i = \text{BIT}([X_{new}, Y_{new}], \mathcal{D}_d^{i-1})) \tag{6}$$

where subscript i refers to the iteration rather than time.

C. Convergence of the Learning Process

The learning procedure converges when the number of correct detections output by $\mathcal{D}_d$ reaches a steady state:

$$\lim_{I \to \infty} \frac{stability_I}{I} = 0 \tag{7}$$

where I is the total number of re-training iterations. In our system, the stability at iteration i is defined as follows:

$$stability(I) = \sum_{i=1}^{I} \|u_i - u_{i+1}\| \tag{8}$$

$$\begin{aligned} u_i \;=\; & \sum_{N_i} y_i \cdot \psi(P(y|X_{valid}^i, \mathcal{D}_d^i)) \\ & + (1 - y_i) \cdot \psi(1 - P(y|X_{valid}^i, \mathcal{D}_d^i)) \end{aligned} \tag{9}$$

where u_i is the number of correct classified examples, X_{valid} is the validation set, ψ is binary function via thresholding 0.5. One can halt the learning process when the stability stops increasing, or other stopping conditions (e.g. number of iterations) are triggered.

IV. APPLICATION TO MULTISENSOR HUMAN DETECTION

In this section, we present an online transfer learning for human classification in 3D LiDAR scans using the robot shown in Fig. 1. The sensor configuration resembles the one adopted for an industrial floor washing robot developed by the EU project FLOBOT[1] and, besides a 3D LiDAR on the top, includes an RGB-D camera and a 2D LiDAR mounted on the front. We describe in the following paragraphs how to use the state-of-the-art detectors to train a 3D LiDAR-based human detector online, instead of training it offline using manually-labeled samples.

The detailed block diagram of our implementation can be seen in Fig. 3. At each iteration, 3D LiDAR scans are first segmented into point clusters. The 2D position and velocity of these clusters are estimated in real-time by a multitarget tracking system, which outputs the trajectories of all the clusters. At the same time, a classifier is trained to classify the clusters as human or not, assigning a normalized confidence value to each of them. This confidence is the predictive probability $P(y_i|x_i, d_j)$ for the 3D LiDAR-based detector, which is needed for the calculation of the trajectory probability in Eq.1-3. The classifier is initialized and retrained online. The trajectories and the probabilities are sent to a label generator, which generates the training labels for the next iteration.

The upper-body detector [10] and the leg detector [7], respectively based on the RGB-D camera and the 2D LiDAR, are the static detectors $\mathcal{D}_s$. Both enable human tracking by sending the position of the detections. In addition, they provide the corresponding probabilities $P(y_i|x_i, d_j)$ (i.e. normalized detection confidence) to the label generator. The combination of 3D-LiDAR-based cluster detector and the

[1] http://www.flobot.eu

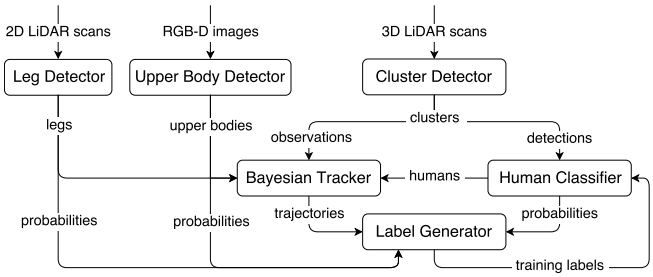

Fig. 3. Process details of the online transfer learning for human detection.

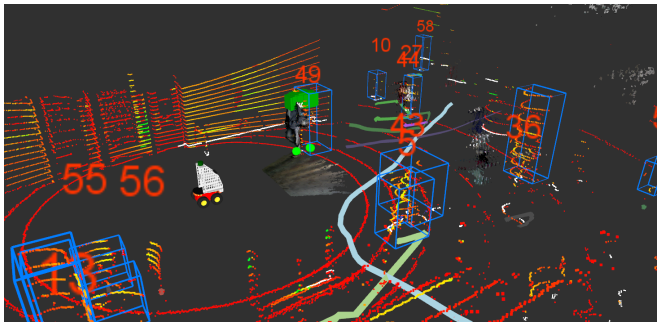

Fig. 4. A screenshot of our multisensor-based detection and tracking system in action. The sparse colored dots represent the laser beams with reflected intensity from the 3D LiDAR. The white dots indicate the laser beams from the 2D LiDAR. The colored point clouds are RGB images projected on depth data of the RGB-D camera. The robot is at the center of the 3D LiDAR beam rings. The numbers are the tracking IDs and the colored lines represent the people trajectories generated by the tracker. For example, the person with tracking ID 49 has been detected by the RGB-D upper-body detector (green cube), the 2D LiDAR leg detector (green circle), and the 3D LiDAR cluster detector (blue bounding box).

human classifier, instead, constitutes the dynamic detector $\mathcal{D}_d$ that we want retrain online. For an intuitive understanding of the various detectors and their outputs, please refer to the example in Fig. 4. The following paragraphs describe each module in detail.

A. Upper Body Detector and Leg Detector

The upper-body detector identifies upper-bodies (shoulders and head) in 2D range (depth) images, taking advantage of a pre-defined template. The confidence of the detection is inversely proportional to the observation range. The leg detector detects legs in 2D LiDAR scans based on 14 features, including the number of beams, circularity, radius, mean curvature, mean speed, and more. Its detection performance is limited by moving and crowd people. As the upper-body detector and leg detector are not probabilistic methods and for the sake of simplifying mathematical conversion, a probability of 0.5 is assigned if x_i is detected as a human.

B. Cluster Detector and Human Classifier

The 3D LiDAR-based cluster detector and the human classifier are originally from our recent work [12], while the former has been incorporated in different problems [23], [24]. As input of this module, a 3D LiDAR scan is first properly segmented into different clusters using an adaptive clustering approach. The latter enables to use different optimal thresholds for point cloud clustering according to the scan ranges. Then, a Support Vector Machine (SVM)-based classifier [25] with six features (a total of 61 dimensions) is trained online. These features are selected to meet robots' requirements for real-time and online computing performance. For more implementation details, please refer to [12].

In our approach (based on LIBSVM [26]), the uncalibrated error function of SVM is squashed into a logistic function (here is the sigmoid function) to get the predictive probability $P(y_i|x_i, d_j)$ (used in Eq. 3). To be more specific, a binary classifier (i.e. human or non-human) is trained at each iteration. The ratio of positive to negative training samples is set to $1:1$, and all data are scaled to $[-1, 1]$, generating probability outputs and using a Gaussian Radial Basis Function kernel [27]. Since LIBSVM does not currently support incremental learning, the system stores all the training samples accumulated from the beginning and retrains the entire classifier at each new iteration. The solution framework, however, also allows for other classifiers and learning algorithms.

C. Bayesian Tracker

People tracking is performed by a robust multisensor-multitarget Bayesian tracker, which has been widely used and described in previous works [2], [8], [12], [17], [28]. The estimation consists of two steps. In the first step, a constant velocity model is used to predict the target state at time t given the previous state at $t-1$. In the second step, if one or more new observations are available from the detectors, the predicted states are updated using a Cartesian or a Polar observation model, depending on the type of the sensor. An efficient implementation of Nearest Neighbor data association is finally included to resolve ambiguities and assign each person the correct detections, in case more than one are simultaneously generated by the same or different sensors. As reported in [2], [17], [28], our tracker fulfills the requirements listed in Sec. III-A. We refer the reader to these publications for further details.

D. Label Generator

The positive training labels X^+ are generated according to Eq. 4, while the negatives X^- are generated based on a volume filter:

$$\begin{aligned} X^- \;=\; \{x_i \mid & \mathrm{W}_i < 0.2,\ \mathrm{W}_i > 1.0, \\ & \mathrm{D}_i < 0.2,\ \mathrm{D}_i > 1.0,\ \mathrm{H}_i < 0.2,\ \mathrm{H}_i > 2.0\} \end{aligned} \tag{10}$$

where W_i, D_i, H_i are the width, depth and height of a 3D cluster x_i. The idea is that clusters without a pre-defined human-like volumetric model will be considered as negative samples for the next training iteration. In our application, the dynamic classifier was trained from scratch without any manually-labeled initial sets. As the validation set is not available, the maximum number of iterations was used as halting criteria.

V. EVALUATION

A. Dataset

We evaluated our framework on a real-world dataset[2] collected in an indoor public area by the robot shown in Fig. 1. The robot was running the Robot Operating System (ROS) [29] and it was manually driven with a gamepad. Several *rosbag* files were recorded, and the total length of which is about 49 minutes, including two continuous recordings of 19 and 30 minutes. Sensor data were recorded in their original frame of reference and the coordinate transformations were handled by the ROS *tf* package.

B. Experimental Setup

The experiments were conducted on the 19 minutes segment of continuous data, in which our binary SVM human classifier was learned online. The classifier was retrained once every 300 new positive (human) and 300 new negative (non-human) samples, labeled by the label generator, corresponding to one iteration. We report the results for the first seven iterations, collecting a total of 2,100 positive and 2,100 negative. In addition, a classifier was trained offline, using 2,100 manually labeled positive samples with an equal amount of randomly selected negative samples, to serve as a baseline for comparison. Furthermore, we arbitrarily selected 100 scan frames from the dataset and fully annotated these (including standing and sitting people) as a test set. This contains 1,197 human labels with varying distances from the robot between few centimeters and twenty meters. A detection was considered a true positive if the overlap between it and the ground truth was more than 50%.

Our framework has been fully implemented within a modular ROS architecture. All components are ready for download[3] and use by other researchers. The dataset collection and all the experiments reported in this paper were carried out on the robot embedded PC, with an Intel i7-4785T processor and 8 GB memory, using Linux Ubuntu 14.04 LTS (64-bit) and ROS Indigo. It is worth noting that our system is fast and cost effective, since it can learn a human detector within minutes and using only inexpensive CPUs, rather than training for hours or days with expensive GPUs.

C. Human Classification

We first evaluate the performance of the 3D LiDAR-based human classification after every online training iteration. We compare the results for all the possible sensor combinations: 3D LiDAR only, i.e. without any knowledge transfer but learned from trajectories only; 3D LiDAR with RGB-D camera; 3D LiDAR with 2D LiDAR; 3D LiDAR with both RGB-D camera and 2D LiDAR. We measure the average precision (AP) [30], rather than the classification accuracy (ACC) used in other methods [13], because more informative. Indeed the number of true negatives in our binary classification was far larger than the number of true positives, leading to an ACC always higher than 80% (with probability threshold $\sigma_t = 0.5$), for each training iteration and each sensor combination. In all the experiments, the trajectory probability threshold σ_t of Eq. 4 was set to 0.7.

[2]https://lcas.lincoln.ac.uk/wp/research/data-sets-software/l-cas-multisensor-people-dataset/

[3]https://github.com/LCAS/online_learning/tree/multisensor

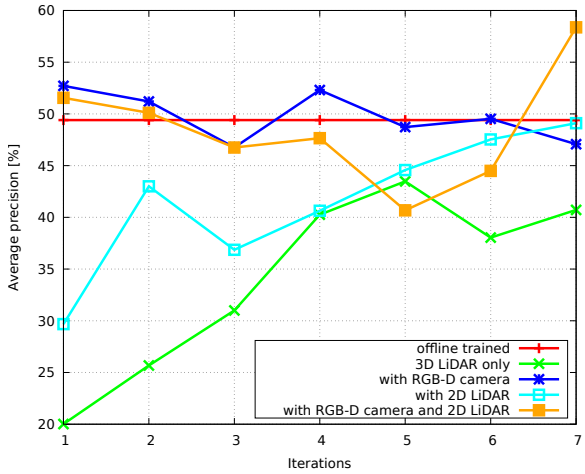

Fig. 5. Comparison of different 3D LiDAR-based human classifiers trained offline (red line) and online using different sensor combinations. Best viewed in color.

The experimental results in Fig. 5, show that the AP for "3D LiDAR only" and (3D LiDAR) "with 2D LiDAR" increases with the iterations, while there are not significant changes "with RGB-D camera". However, the results "with RGB-D camera and 2D LiDAR" show an interesting trend, first decreasing until the 5th iteration, and then increasing well above any other combination. This final outcome shows the advantage of our multisensor system, which can eventually improve the online transfer learning process.

We additionally evaluate the Precision-Recall of the offline and final (i.e. after the 7th iteration) online trained classifiers. The experimental results are shown in Fig. 6. Once again, the combination of RGB-D camera and 2D LiDAR achieves the best performance. Differently from the "3D LiDAR only" case, which learned only from moving people, the solution "with RGB-D camera and 2D LiDAR" was able to learn from moving, standing and sitting people, which greatly improved the human classification performance. It is also worth pointing out that, despite showing a relatively high precision, the true positive rate (recall) of the offline trained classifier is generally lower. This is due to a lack of long-distance samples in the offline training set, which are difficult to label by a human annotator.

VI. CONCLUSION

In this paper, we presented a framework for online transfer learning, applied to 3D LiDAR-based human classification, taking advantage of multisensor-based tracking. The framework, which relies on the computation of human trajectory probabilities, enables a robot to learn a new human classifier over time with the help of existing human detectors. To this end, we proposed a semi-supervised learning method, which fuses both model-based (labeled) and model-free (unlabeled)

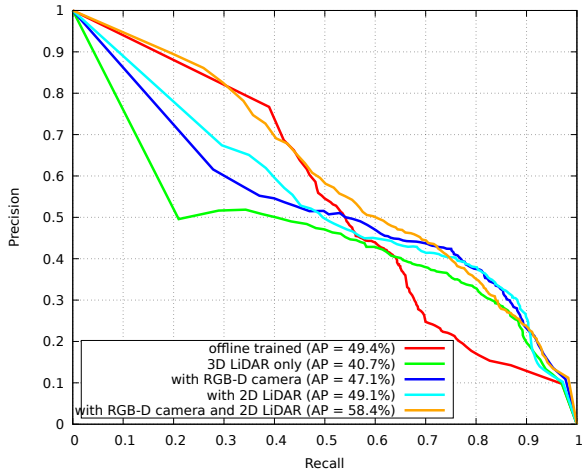

Fig. 6. Performance evaluation of human detection. Best viewed in color.

detections from different sensors. A very promising feature of the proposed solution is that the new human classifier can be learned directly from the deployment environment, thus removing the dependence on pre-annotated data. The experimental results, based on a real-world dataset, demonstrated the efficiency of our system.

The proposed framework has been fully implemented into ROS with a high level of modularity. The software and the dataset are publicly available to the research community, with the intention to perform objective and systematic comparisons between the recognition capabilities of different robots. Moreover, our framework is easy to extend to other sensors and moving objects, such as cars, bicycles and animals.

Despite these encouraging results, there are several aspects which could be improved. For example, the AP of the online learned classifier is still relatively low, due to the complexity of the environment recorded in our dataset. This can be further improved by using a more advanced model for negative sample generation. In addition, it remains to be verified how a new human detector, based on the online trained classifier, will affect the stability of the system and its tracking performance.

ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 645376 (FLOBOT) and No. 732737 (ILIAD).

REFERENCES

[1] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *International Journal of Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.

[2] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," in *Proc. of ICRA*, 2016, pp. 5512–5519.

[3] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia, "People tracking and following with mobile robot using an omnidirectional camera and a laser," in *Proc. of ICRA*, 2006, pp. 557–562.

[4] D. Held, J. Levinson, and S. Thrun, "Precision tracking with sparse 3d and dense color 2d data," in *Proc. of ICRA*, 2013, pp. 1138–1145.

[5] K. Misu and J. Miura, "Specific person detection and tracking by a mobile robot using 3D LIDAR and ESPAR antenna," in *Proceedings of IAS*, 2012, pp. 705–719.

[6] K. Koide and J. Miura, "Person identification based on the matching of foot strike timings obtained by LRFs and a smartphone," in *Proc. of IROS*, 2016, pp. 4187–4192.

[7] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Proc. of ICRA*, 2007, pp. 3402–3407.

[8] N. Bellotto and H. Hu, "Multisensor-based human detection and tracking for mobile service robots," *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, vol. 39, no. 1, pp. 167–181, 2009.

[9] M. Munaro and E. Menegatti, "Fast RGB-D people tracking for service robots," *Autonomous Robots*, vol. 37, pp. 227–242, 2014.

[10] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *Proc. of ICRA*, 2014, pp. 5636–5643.

[11] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. Morgan & Claypool, 2009.

[12] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for human classification in 3d lidar-based tracking," in *Proc. of IROS*, 2017, pp. 864–871.

[13] A. Teichman and S. Thrun, "Tracking-based semi-supervised learning," *International Journal of Robotics Research*, vol. 31, no. 7, pp. 804–818, 2012.

[14] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: principles and techniques*. Storrs, CT: University of Connecticut, 1995.

[15] Z. Yan, N. Jouandeau, and A. Ali Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 399, December 2013.

[16] L. Spinello, R. Triebel, and R. Siegwart, "Multiclass multimodal detection and tracking in urban environments," *International Journal of Robotics Research*, vol. 29, no. 2, pp. 1498–1515, 2010.

[17] C. Dondrup, N. Bellotto, F. Jovan, and M. Hanheide, "Real-time multisensor people tracking for human-robot spatial interaction," in *ICRA Workshop on Machine Learning for Social Robotics*, 2015.

[18] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining RGB and dense LIDAR data," in *Proc. of IROS*, 2014, pp. 4112–4117.

[19] A. González, G. Villalonga, J. Xu, D. Vázquez, J. Amores, and A. M. López, "Multiview random forest of local experts combining RGB and LIDAR data for pedestrian detection," in *Proc. of IV*, 2015, pp. 356–361.

[20] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proc. of ICRA*, 2000, pp. 476–481.

[21] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI magazine*, vol. 9, no. 2, p. 61, 1988.

[22] J. Read, A. Bifet, B. Pfahringer, and G. Holmes, "Batch-incremental versus instance-incremental learning in dynamic and evolving data," in *Proc. of IDA*, 2012, pp. 313–323.

[23] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett, "3DOF pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data," in *In Proceedings of ICRA*, Brisbane, Australia, May 2018.

[24] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett, "Recurrent-octomap: Learning state-based map refinement for long-term semantic mapping with 3d-lidar data," *IEEE Robotics and Automation Letters*, 2018.

[25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.

[27] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, 2003.

[28] 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, 2010.

[29] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[30] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.