Scene in the Loop: Towards Adaptation-by-Tracking in RGB-D Data

Luciano Spinello, Cyrill Stachniss and Wolfram Burgard
University of Freiburg, Germany

***Abstract*—This paper addresses the problem of adapting an existing object detector to the characteristics of the environment in an unsupervised manner. The technique aims to reject all the false positive detections by exploiting the information from the environment and from the tracking system. We follow the intuition that similar characteristics are shared among the objects that are present in the same scene. Our aim is to detect the false positives by analyzing which detections do not share common properties in RGB-D feature space. For this, we make use of a One-class SVM in an unsupervised manner. This idea allows our approach to adapt to the environment it is tracking in. We developed and evaluated our system based on a people detection and tracking system that operates on Kinect data. Our experimental evaluation shows that our method outperforms standard outlier detection techniques and that is able to remove over 50% of the false positives without eliminating a significant amount of correct detections.**

I. Introduction

Tracking objects in the environment from image and depth data is a key task for several robotics applications. To track objects of interest, most approach use object detection techniques that are trained to respond to specific categories of objects in the sensory data. Prominent examples are cars or pedestrians in street scenes [1, 2, 18] or everyday objects with which a service robot has to interact in a household environment. Typically, object detectors are trained beforehand by using large datasets. During operation, the detectors recognize the objects and the tracking module takes care of estimating the position of the object in the environment. A key problem when applying such processing pipeline is that detection is never perfect. It is practically impossible to build a detector that is error-free in every environment and noise condition.

Great efforts have recently been taken for collecting datasets that present objects with large appearance variety and different backgrounds [5, 6]. Despite the efforts to robustify the generalization capabilities of object detection methods, datasets appear to have a strong build-in bias so that a classifier trained on one datasets underperforms in others [20], the so called 'dataset-bias' effect. This especially happens if the environment that has been used for training has significantly different appearance characteristics with respect to the testing environment. A different sensitivity to shadows, structures producing strong gradients, light conditions and optical quality of the imagery can generate unwanted detection responses and produce false positive detections, see Figure 2.

In this paper, we consider the problem of adapting an existing object detector to the characteristics of the environment in order to obtain lower false positive rates. We follow the intuition that similar appearance/depth characteristics are shared among all the objects that are present in the same scene.

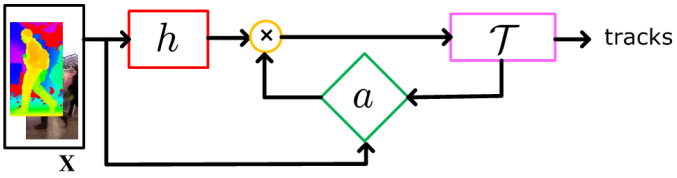

Fig. 1. Architecture of our method. $\mathbf{X}$ is the RGB-D data frame, $h(\cdot)$ is the object classifier, $a(\cdot)$ is the adaptation function, and $\mathcal{T}$ is the tracking module. In this paper, the adaptation function collects features computed in RGB-D data to learn a One-class SVN model for rejecting false positive detection in an unsupervised manner.

Our contribution consists in formulating an object detection adaptation system that exploits the information coming from the tracking module, in an unsupervised manner. Our approach is formulated as a one-class classification problem and solved via a One-class SVM [15]. This machine learning technique is used to learn a feature model of the "real objects" by extracting RGB-D features from all the tracked targets, that usually include correct detections (inliers) and false positives (outliers).

The approach is unsupervised in the sense that the system does not need any labeling information from the user but it only requires a coarse estimate of the ratio between outliers and inliers. Our approach exploits the tracking information by considering as valid data for adaptation only the object detections that are consistent over time (such that they are successfully included in a track). This simplifies the feature space and the adaptation capability.

To the best of our knowledge, our system is the only one that is able to address and remove systematic false positive detections that might generate entirely false tracks. These cases are usually the hardest to remove because they are generated by specific environment configurations such as strong gradients, cluttered areas or unexpected light conditions. Moreover, it is the first time that a tracking system is used for improving a *generic* object detection performance in an unsupervised manner and it is the first time that a One-class SVM technique is used in this context. Our approach can be used in conjunction with most detectors and any tracking approach.

II. Related Work

The problem of semi-supervised object detection and tracking as well as the adaptation of existing classifiers has also been studied by other researchers. For example, Teichman

and Thrun [18] presented a semi-supervised approach to track classification in urban traffic scenes using dense 3D depth data. This approach requires segmented point clouds for the classification of objects and tracking without a class model. It iteratively trains a classifier and extracts useful training examples from unlabeled data by exploiting tracking information. The approach is reported to require only few hand-labeled training tracks of each object class and still provides a competitive performance. The recent approach of Teichman et al. [19] is a semi-supervised learning method that uses tracking information to find new, useful training examples automatically. It aims at classifying the tracks of all visible objects not requiring individual point cloud segments. The authors propose a new track classification method that can be executed online, is not specific to the object class and provides high accuracy for the task of classifying correctly-tracked, well-segmented objects into car, pedestrian, bicyclist and background classes.

The problem of adapting object detection system to the current scene received considerable attention recently and approach, for example, for category models to new domains [14]. In this work, Saenko et al. propose a method for transferring models from labeled datasets acquired in one image domain to other environments. This can be used to account, for example, for different cameras used in training and testing phase. The key idea is to learn a metric that compensates for the transformation of the object representation that occurred due to the domain shift. Breitenstein et al. [2] proposed a tracking-by-detection approach in a particle filtering framework that uses the confidence of the detectors and online trained instance-specific classifiers as a graded observation model. Thus, generic object category knowledge is complemented by instance-specific information. A central contribution of their work is the investigation of how unreliable information sources can be used for multi-person tracking. Recent works focus in detecting objects in RGB-D data [10, 16]. Both approaches have been shown to be robust to real-world Kinect data collected in indoor environments. As most detection approaches, the systems are not free of false detections. Therefore, our current work aims to adjust the tacking-by-detection process to be more reliable by eliminating false positive detections. Our current approach is also different to [17], which focuses on adaptive fusion and domain adaptation for object detection in RGB-D data. Their approach works in a totally supervised manner and primary focuses on adaptive fusion scheme for RGB and D cues.

III. Overview

This section describes our main contribution. Please see Fig. 1 for an overview of our architecture. The input of the system are RGB-D data frames. The processing pipeline is composed of a detector trained for an object category and a tracking system that computes estimated trajectories of the detected objects in a scene. We use the following notation in the paper: given a RGB-D data frame $\mathbf{X}$, we indicate the detector as $h(\mathbf{X})$ and the tracking system as $\mathcal{T}$. In addition to this standard tracking architecture, we included an adaptation block $a(\mathbf{X}, \mathcal{T})$ that gates the output of $h(\mathbf{X})$ in order to modify the input of $\mathcal{T}$. The tracks and the RGB-D data is fed back to the adaptation block $a(\cdot)$ that uses this data over time to learn which detector output to discard.

Fig. 2. Visual exemplification of the 'dataset bias'. The left is a RGB sample image from the training set 'TUD-Brussels'[21], the right one is an image taken from the test dataset. Separate RGB and depth imagery are used for training the RGB-D detector. The RGB training set collects images of urban outdoor scenarios, where shadows, clutter and lighting conditions make it very different from the test scenario, that has been acquired indoor in a university hall. The training set has been recorded by using high quality optics, the test set uses budget hardware resulting in less sharp imagery.

Object tracking is a very well studied field. Several successful techniques have been presented in the recent years [2, 7, 9]. Our method follows a *tracking-by-detection* approach: the tracker takes as input the output of a detector trained for a certain object category. The input can be binary or real-valued. The object detector is trained in a standard supervised manner with a dataset containing many positive and negative examples.

Our aim is to fight the 'dataset-bias' and to build an object detection system that adapts to the current scene. Our approach makes use of object tracks, i.e. filtered detections that are consistent over time, for collecting object information in the environment. We follow the intuition that similar appearance/depth characteristics are shared among the objects that occur in the same scene. Thus, the tracks generated by false objects (false positives) potentially have different appearance (in RGB-D sense) with respect to the real ones. In practice, our architecture computes an object detector domain adaptation: the response of the detection system is changed by exploiting characteristics of the objects present in the environment. Here, it is important to notice that the proposed architecture is orthogonal to the choice of the tracker and to the kind of detector, as soon as the latter is able to achieve high recall rate. This is important in order to not miss relevant objects in the scene in the early detection step.

Notice that this work differs from standard target adaptation methods that instead aim to build, highly specific classifiers for each tracked target (e.g,: [9]), in a online fashion. In those works, no knowledge is shared between the various classifiers and no rejection of false positives is possible. Our work approaches the problem of adaptation in a different way:

the goal is to find a common description of the appearance that is shared between all the true objects in the environment, in an unsupervised manner. To achieve this goal, we build $a(\cdot)$ with a machine learning technique based on a One-class SVM that is able to efficiently work in an unsupervised manner with high dimensional features and lots of data.

IV. RGB-D Detector

The proposed method is general with respect to the choice of the object detector. In this work, we employ the recently proposed RGB-D Combo-HOD detector with adaptive sensor fusion [17] that combines the outputs of an Histogram of Oriented Gradients (HOG) detector for images with an Histogram of Oriented Depths (HOD) detector for dense depth data and fuses the two modalities in an adaptive manner. As stated before, our approach is orthogonal to the detector itself, assuming that it provides high recall rates.

The detector $h(\cdot)$ takes as input a portion of a RGB-D data frame $\hat{\mathbf{X}} \subset \mathbf{X}$ and computes a confidence about the presence of an object in it. It is often possible to model the detector output $h(\hat{\mathbf{X}})$ as a probability[4, 13]:

$$p(\pi \,|\, \hat{\mathbf{X}}, \theta_1) \;\simeq\; r(\, h(\hat{\mathbf{X}})\,) \tag{1}$$

where $r(\cdot)$ is the function that maps the detector output to probabilities (e.g. a sigmoid), θ_1 are the parameters learned for the classifier $h(\cdot)$ and π indicates the existence of an object in $\hat{\mathbf{X}}$. The detector runs at several locations in the RGB-D data frame. For notation simplicity, we use $h(\mathbf{X})$ to indicate all the detections computed for the entire RGB-D frame $\mathbf{X}$ by the object detector.

V. Adaptation-by-Tracking

This section presents our method to formulate and compute the adaptation mechanism for object detection in an unsupervised manner.

The idea is to build a function $a(\cdot)$ which by inspecting the RGB-D data associated with the current tracks (i.e. bounding boxes) is able to reject false positives in the upcoming frames. The tracking module plays a key role in the adaptation mechanism. Tracking can be seen as a first filtering stage for occasional false detection responses, triggered for example by temporary noise or clutter. Such false detections will not be included as tracks due to weak space-time consistency. This fact is exploited by our approach. The adaptation function is learned using only data that is associated with tracks. Thus, the input data for learning $a(\cdot)$ is potentially polluted by a relatively small quantity of false positives. It therefore has the potential to represent the "true objects" well.

The idea of the adaptation function shares some common ground with the Mixtures of Local Experts method developed by Jacobs et al. [8]. In that work, the authors fuse the response of several classifiers by multiplying their output with gating functions that use the classifiers input. In our case, there is only one gating function, $a(\cdot)$, and it is used to scale the detector confidence, $h(\mathbf{X})$, see Figure 1.

The online adaptation mechanism follows two distinct phases: the boostrapping and the gating phase. The bootstrapping phase is used for learning $a(\cdot)$ and lasts for an amount of time fixed beforehand by the user. During the gating phase, $a(\cdot)$ is used for filtering detections. The adaptation system can be reset and re-bootstraped after a certain amount of time or when the scene appearance changes significantly.

A. Bootstrapping the Adaptation Function

The first phase consists in collecting data for training $a(\cdot)$. This is achieved by processing the output of the tracking module during operation. The tracking module $\mathcal{T}$ estimates at each time step the position of N targets and computes their position in the RGB-D data frame:

$$\begin{aligned} &\mathbf{l}_i = [x, y] && \mathbf{b}_i(\mathbf{l}_i) = [x_I, y_I, w_I, h_I] \\ &\mathcal{L} = \{\mathbf{l}_1, ..., \mathbf{l}_N\} && \mathcal{B} = \{\mathbf{b}_1, ..., \mathbf{b}_N\} \\ &\hat{\mathcal{X}} = \{\hat{\mathbf{X}}_{\mathbf{b}_1}, ..., \hat{\mathbf{X}}_{\mathbf{b}_N}\} \end{aligned} \tag{2}$$

where $\mathbf{l}_i$ is the estimated 2D position of a target in world coordinate frame, $\mathbf{b}_i(\mathbf{l}_i)$ is its bounding box in the RGB-D data frame and $\hat{\mathbf{X}}_{\mathbf{b}_1}$ is the RGB-D data enclosed in the bounding box. We then compute shape-describing RGB-D features for each element of $\hat{\mathcal{X}}$:

$$\begin{aligned} &\mathbf{f}(\hat{\mathbf{X}}_{\mathbf{b}_i}) \in \mathbb{R}^D \qquad \hat{\mathbf{X}}_{\mathbf{b}_1} \subset \hat{\mathcal{X}} \\ &\mathcal{F} = \{\mathbf{f}_1, .., \mathbf{f}_N\} \end{aligned} \tag{3}$$

Here, $\mathbf{f}$ consists of a coarsely described HOG descriptor in depth and RGB data and $D = 64$. In the bootstrapping phase, a fixed amount of $\mathcal{F}$ data is collected by letting the tracker run for several time steps, $\mathcal{U} = \{\mathcal{F}_1, ..., \mathcal{F}_M\}$. The set $\mathcal{U}$ is a feature-based appearance description of all the targets followed by the tracker in the bootstrap phase. The data in $\mathcal{U}$ is likely to contain false detections (false positives) but it is assumed to contain a majority of true positives. The intuition is that the features describing the false positives will be apart from the features of the real objects. We are interested in detecting these outliers. We aim to compute a probability distribution of the features that describe the true objects in order to compute outlier likelihoods. The problem is that our input is unsupervised thus it has unknown labels (true positive or false positive) and contains feature noise.

Several works have been presented that address the problem of outlier detection, for example [3, 11]. As a straight forward approach, clustering techniques with heuristics based on nearest neighbors could be used to achieve this goal. Such approaches, however, typically show suboptimal performances when applied with robust, high-dimension features. In our approach, the high dimensionality of the feature space and its sparsity makes the problem hard to solve with simplistic techniques. Thus, we propose to solve this problem by formulating it as a one-class classification problem and solving it via a One-class SVM method. We learn the adaptation function a as a one-class model $\mathcal{S}$ that is trained by using the collected unlabeled features $\mathcal{U}$. A brief discussion and the formulation of the One-class SVM approach is presented in the Section V-C.

In the bootstrap phase, $a(\cdot) := 1$ and adaptation is not active.

B. Gating with the Adapation Function

In the second phase of the algorithm, the learned one-class classification model is directly used for computing the adaptation function $a(\cdot)$ and it is not further updated. In the current implementation, we use only the binary output of the one-class SVM:

$$a(\mathbf{X}, \mathcal{T}) = \mathcal{S}(\mathbf{f}, \theta_2) = \{0, 1\} \tag{4}$$

where $\mathcal{S}$ is the one-class SVM function, and θ_2 are its learned parameters. As soon as all the detections in a RGB-D frame are computed by $h(\cdot)$, the associated bounding boxes are used for computing features $\mathbf{f}$ then the learned SVM classifies this data as outlier or not. Essentially, for each detection $h(\hat{\mathbf{X}})$ in the RGB-D frame, the tracker input becomes:

$$a(\hat{\mathbf{X}}, \mathcal{T}) \cdot h(\hat{\mathbf{X}}) = \begin{cases} p(\pi \mid \hat{\mathbf{X}}, \theta_1), & \text{if } \hat{\mathbf{X}} \text{ is an inlier} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

C. One-class SVM classification

The standard problem of classification is the problem of identifying the category which a data point belongs to. This decision is taken on the basis of a training set containing data whose category membership is known beforehand.

In one-class classification, a data point is just classified as an inlier or outlier. Moreover, there is a substantial difference between standard classification and one-class classification: in the latter, it is assumed that imperfect information of only a single class is available. This means that no explicit information about the class of outliers is present. The boundary between the two classes has to be estimated from noisy and unlabeled data. The reasoning is that it is often intractable to characterize the distribution of the outliers (false positives) because they belong to an unknown number of different "negative" classes.

Particularly robust methods for addressing one-class classification are One-class SVMs. One-class SVM is an algorithm which computes a binary function that estimates the regions in input space where data can be explained as a probability density function. In short, a One-class SVM tries to capture the support within which the positive examples are located, with the aim to separate them from all the rest. After transforming the data points via a kernel, One-class SVM treats the origin as the only member of the second class (the outlier class). Data is separated from the origin by solving the following quadratic program:

$$\begin{aligned} \min_{w \in F, \xi \in \mathbb{R}^E, \rho \in \mathbb{R}} \quad & \frac{1}{2}\|w\|^2 + \frac{1}{\nu E} \cdot \sum \xi_i - \rho \\ \text{subject to} \quad & w \cdot \Phi(\mathbf{f}_i) > \rho - \xi_i, \xi_i > 0, i \in [1, E] \end{aligned} \tag{6}$$

where E is the total number of feature vectors in $\mathcal{U}$, w and ρ are the margins to optimize, ξ_i is a slack variable (as in the standard soft-margin SVM formulation) and Φ is the kernel function. Two parameters have to be set before the optimization, the kernel Φ and the expected ratio of outliers in the training set $\nu \in (0, 1)$. The parameter ν is critical to avoid overfitting or underfitting. In our case, we have used linear kernel and we have determined ν based on a validation set. By solving the optimization in Equation 6, we have generated a model for object inliers described by features $\mathbf{f}$, denoted as $\mathcal{S}(\mathbf{f}, \theta_2)$.

VI. Experiments

The experiments are designed to show the improvements of our unsupervised approach to learn an adaptation function for reducing the number of false positives detections. For the evaluation of our method, we have chosen people as objects to detect and adapt for the reasons that there exist well-established detectors. Moreover, the people category is one of the most challenging object categories: humans are articulated objects that exhibit a large variability in their appearance due to different body poses, clothing, or wearable luggage.

A. RGB-D Data Set

We make use of the *canteen RGB-D dataset*, a large-scale indoor data set with unscripted behavior of people [12]. The data set consists of $3000+$ frames and it has been recorded in the lobby of a university canteen at lunch time. The data set has been manually annotated to include bounding boxes and the visibility status of subjects (fully visible/partially occluded).

B. Results

We have quantified the advantages of the proposed technique by analyzing standard performance indices. For evaluation, we have used a ground truth track assignment given by the annotated dataset. We additionally added false-positive tracks (tracks entirely composed of false positive detections) and added false-negative misdetections to true-positive tracks.

The first experiment is designed to show the robustness of the adaptation system with respect to the quality of data. For analyzing this aspect, we have evaluated the robustness at different level of noise and mislabeling in the bootstrapping process. In our experimets, we have fixed the length of the bootstrapping process to 500 frames, $\nu = 0.35$ and we have gradually increased the number of false positive tracks, thus the number of false positive detections (outliers). In the plot in Figure 3-left, we evaluate the true positive rate (TPR) and the ratio of discarded false positives (rmFR), computed in the entire dataset, with respect to the ratio of outliers in the bootstrap phase. In practice, the 'x' axis depicts the ratio of outliers in the One-class SVM learning phase. Theoretically optimal results in this graph are: a TPR curve that is constant and close to 1, a rmFR curve constant and close to 1. TPR and rmFR are computed with respect to a detection system without adaptation. This means that the system would be able to reject all false positives meanwhile being able to detect true positives with unchanged ability. Consider the true positive rate curve TPR-RGBD and the discarded false positive rate curve rmFPR-RGBD, respectively the red and blue continuous curves in Figure 3-left. In our case, TPR-RGBD is fairly constant and close to 0.75. This value guarantees good tracking performances (not many targets are lost due to missed detections). In contrast to that, many false positives are rejected: rmFPR-RGBD goes from 0.68 to 0.48. With this rate, many outliers are rejected and potentially many erroneous tracks are not initialized. Especially

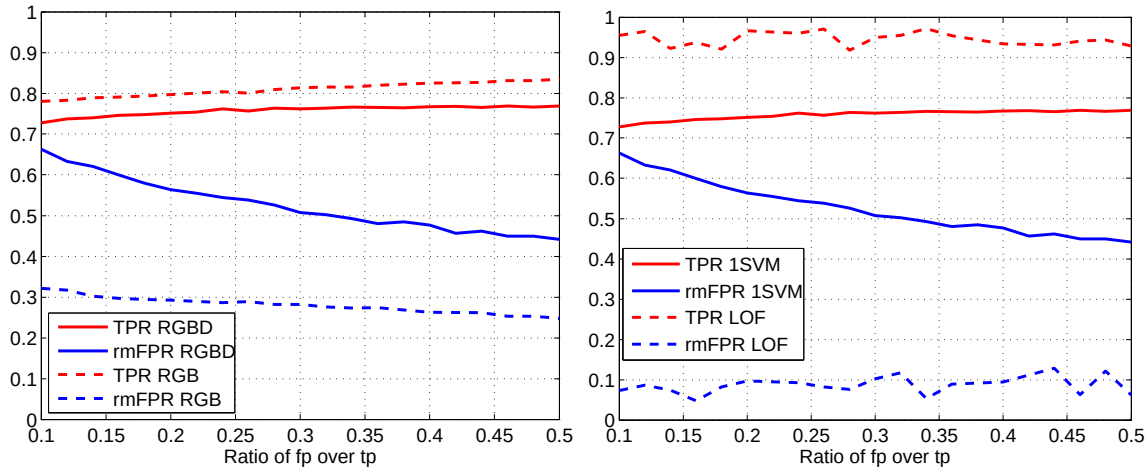

Fig. 3. Robustness of the adaptation system with respect to the quality of data (outlier ratio in the bootstrap phase). True positive rate (TPR) and rejected false positives rate (rmFP) are computed for the entire dataset. **Left:** Comparison of the system using RGB and RGB-D features. When using the full RGB-D data (continuous curves), the system is able to achieve a high detection rate and an high rejection of false positives. When using features computed with RGB-only data (dashed curves), the task of outlier rejection is harder and only a few percentage are discarded.-left. **Right:** Comparison between our system and LOF outlier detector. Our system is depicted with continuous curves, the LOF method with dashed lines. LOF largely underperforms with respect to our approach due to the difficulty of correctly modeling the feature density in a sparse and high dimensional space.

at high error rates, the rmFPR-RGBD curve decreases: the One-class SVM still tries to model a very complicated noisy feature distribution but tends to be more conservative and discards errors with more difficulty.

Another experiment evaluates the improved adaptation capability when RGB-D features are used instead of RGB-only, see Figure 3-left dashed curve. With RGB-only features, the performance is substantially lower, and a low quantity of false positives are removed. Using the additionally depth information alleviates the classification task thus it eases the outlier rejection.

We also compared our technique with the "Local Outlier Rejection" (LOF) method [3], see Figure 3-right. LOF largely underperforms with respect to our approach due to the difficulty of correctly modeling the feature density in a sparse and high dimensional space. For this reason, it is complicated to decide which kind of distance to use and which threshold to use. In our experiments, LOF did not manage to reject more than 10% of the false positives.

Additionally, we evaluated the impact of the adapted detector in the tracking context by using the MOTA index. It is important to notice that false tracks have a high impact on the tracking performance. As soon as the detector is able to reject systematic and consistent false-positive detections, a new track is not initialized and the MOTA index increases. The reduced detection rate on the real object tracks resulting from the adapted detector has less impact on the tracking performances because the tracker is still able to "fill the small gap" in an object's trajectory. In an experiment consisting of a strongly noised bootstrapping phase ($\nu = 0.3$) and a track initialization after 4 consistent consecutive detections, the MOTA index increases of 40%. In a further experiment with a relatively clean bootstrapping phase ($\nu = 0.1$), the MOTA index increases up to 60%.

Figure 4 qualitatively shows false positive tracks, due to systematic and consistent false positive detections, that can be removed with our method.

The system takes approximately $1s$ for computing the One-class SVM model, and after learning, a few milliseconds to classify RGB-D data as outlier or a real object.

VII. Conclusion

In this paper, we propose a novel approach to automatically adapt object detection systems, trained on generic datasets, to the characteristics of scene in order to reduce the number of false positives. Our method collects RGB-D data features in locations where objects are tracked by a tracking system. These features are used to learn a One-class SVM that is able to reject false positive detections. Our approach is unsupervised in the sense that no human needs to label false detections. The model is able to learn an implicit object feature distribution that is robust to noise and mislabeling input. After an unsupervised learning phase, the output of the learned SVM is used to rescale the confidence of the detector.

We have presented experiments in the context of people tracking with a large RGB-D dataset. We have shown the reliability of our approach with respect to high level of outliers in the crucial bootstrap learning phase. We have highlighted the performance increase of using the full RGB-D data instead of using only image data and we have shown that our approach largely outperforms other standard outlier rejection methods such as LOF.

Acknowledgments

This work has partly been supported by the EC under FP7-ICT-248258-First-MM, FP7-ICT-260026-TAPAS, FP7-248873-RADHAR as well as by Microsoft Research, Redmond.

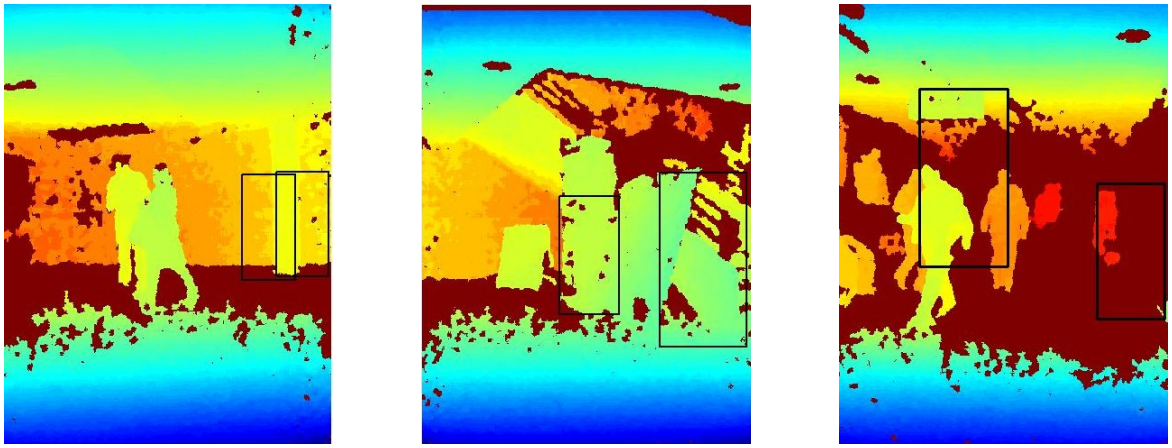

Fig. 4. Three depth views of the environment. Several false positive tracks, depicted by black boxes, that are generated by consecutive false positive detections can be removed with the proposed method

REFERENCES

[1] M Andriluka and B Roth, S. abd Schiele. Monocular 3d pose estimation and tracking by detection. In *IEEE Conf. on Comp. Vis. and Patt. Rec.*, 2010.

[2] M Breitenstein, F Reichlin, B Leibe, E Koller-Meier, and L Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Int. Conf. on Computer Vision*, 2009.

[3] M. M Breunig, H.-P Kriegel, R. T Ng, and J Sander. LOF: identifying density-based local outliers. *SIGMOD*, 29(2), 2000.

[4] T Cover and P Hart. Nearest neighbor pattern classification. *IEEE Tran. on Inf. Theory*, 13:21– 27, 1967.

[5] M Everingham, L Van Gool, C. K. I Williams, J Winn, and A Zisserman. The PASCAL Visual Object Classes Challenge Results (VOC2011), 2011.

[6] G Griffin, A Holub, and P Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[7] E Horbert, K Rematas, and B Leibe. In *Int. Conf. on Computer Vision*, pages 1871 –1878, 2011.

[8] R Jacobs, M Jordan, S Nowlan, and G Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 3:79–87, 1991.

[9] C.-H Kuo and R Nevatia. How does person identity recognition help multi-person tracking? In *IEEE Conf. on Comp. Vis. and Patt. Rec.*, pages 1217–1224, 2011.

[10] K Lai, L Bo, X Ren, and D Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *IEEE Int. Conf. on Rob. & Aut.*, 2011.

[11] A Lazarevic and V Kumar. Feature bagging for outlier detection. In *ACM SIGKDD int. conf. on Knowledge discovery in data mining*, 2005.

[12] M Luber, L Spinello, and K. O Arras. People tracking in rgb-d data with on-line boosted target models. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2011.

[13] J. C Platt. Probabilities for SV Machines. *Advances in Large-Margin Classifiers*, pages 61–74, 2000.

[14] K Saenko, B Kulis, M Fritz, and T Darrell. Transferring visual category models to new domains. Technical Report UCB/EECS-2010-54, EECS Department, University of California, Berkeley, May 2010.

[15] B Schölkopf, J. C Platt, J. C Shawe-Taylor, A. J Smola, and R. C Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, July 2001.

[16] L Spinello and K. O Arras. People detection in RGB-D data. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2011.

[17] L Spinello and K. O Arras. Leveraging RGB-D data: Adaptive fusion and domain adaptation for object detection. In *IEEE Int. Conf. on Rob. & Aut.*, 2012.

[18] A Teichman and S Thrun. Tracking-based semi-supervised learning. In *Proc. of Robotics: Science and Systems*, 2011.

[19] A Teichman, J Levinson, and S Thrun. Towards 3d object recognition via classification of arbitrary object tracks. In *IEEE Int. Conf. on Rob. & Aut.*, 2011.

[20] A Torralba and A Efros. Unbiased look at dataset bias. In *IEEE Conf. on Comp. Vis. and Patt. Rec.*, 2011.

[21] C Wojek, S Walk, and B Schiele. Multi-cue onboard pedestrian detection. In *IEEE Conf. on Comp. Vis. and Patt. Rec.*, 2009.