2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Tracking People in 3D Using a Bottom-Up Top-Down Detector

Luciano Spinello, Matthias Luber and Kai O. Arras

Social Robotics Lab, University of Freiburg, Germany
{spinello, luber, arras}@informatik.uni-freiburg.de

Abstract— **People detection and tracking is a key component for robots and autonomous vehicles in human environments. While prior work mainly employed image or 2D range data for this task, in this paper, we address the problem using 3D range data. In our approach, a top-down classifier selects hypotheses from a bottom-up detector, both based on sets of boosted features. The bottom-up detector learns a layered person model from a bank of specialized classifiers for different height levels of people that collectively vote into a continuous space. Modes in this space represent detection candidates that each postulate a segmentation hypothesis of the data. In the top-down step, the candidates are classified using features that are computed in voxels of a boosted volume tessellation. We learn the optimal volume tessellation as it enables the method to stably deal with sparsely sampled and articulated objects. We then combine the detector with tracking in 3D for which we take a multi-target multi-hypothesis tracking approach. The method neither needs a ground plane assumption nor relies on background learning.**

The results from experiments in populated urban environments demonstrate 3D tracking and highly robust people detection up to 20 m with equal error rates of at least 93%.

I. Introduction

People detection and tracking is a key skill for mobile robots and intelligent cars in populated environments. While most of the related work in this area used vision for this task, range sensing is a particularly interesting sensor modality due to its accuracy, large field of view and robustness with respect to illumination changes and vibrations, the latter points being of particular relevance for mobile observers.

In this paper we address two problems, detecting people in 3D range data and tracking people in 3D space. We extend our previous work on 3D people detection [1] by the tracking stage and an additional top-down procedure in the detection pipeline. This procedure aims at reducing false positives that typically occur with sparsely sampled individuals at large distances from the sensor. We further combine detection with tracking and present results from a tracker this is able to estimate the motion state of multiple people in 3D. To this end, we employ a multi-hypothesis tracking approach (MHT) by Reid [2] and Cox *et al.* [3]. In the experiments we compare our approach with related techniques for detection in 3D range data, in particular spin images [4] and template-based classification.

While there is little related work for people detection and tracking in 3D, many researchers addressed this task using 2D range data. In early works [5], [6], [7], people are detected using ad-hoc classifiers, looking for moving local minima in the scan. The first principled learning approach has been taken by Arras *et al.* [8] where a classifier for 2D point clouds has been learned by boosting a set of geometric and statistical features. As there is a natural performance limit when using only a single layer of 2D range data, several authors have been using multiple co-planar 2D laser scanners [9], [10], [11]. In the field of people detection in 3D data, Navarro *et al.* [12] collapse the 3D scan into a virtual 2D slice to find salient vertical objects above ground. They align a window to the principal data direction, compute a set of features, and classify pedestrians using a set of SVMs. Bajracharya *et al.* [13] detect people in point clouds from stereo vision by processing vertical objects and considering a set of geometrical and statistical features of the cloud based on a fixed pedestrian model. Both of them can be seen as top-down detection procedures.

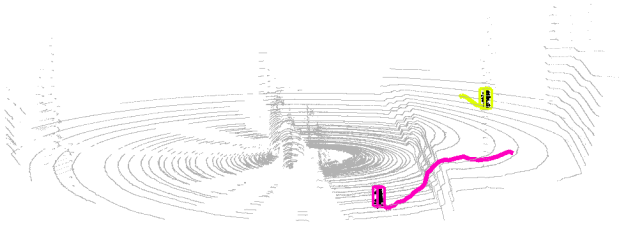

Fig. 1. Two persons are detected and tracked in 3D, one is descending a flight of stairs. The z axis in the figure is magnified for clarity.

Our approach extends the state-of-the-art in several aspects. First, all these works require a ground plane assumption which prevents them from being true 3D methods. This is not required in our case. We combine a top-down classifier with a bottom-up detector and quantify the improvement of this combination. We then compare our method to two established 3D object detection techniques, spin images and template-based classification, and demonstrate that with equal error rates (EER) of at least 93% for people up to 20 m afar, our approach clearly outperforms the alternative techniques. Finally, we combine our detector with 3D tracking.

The paper is structured as follows: the combined bottom-up top-down detection approach is presented in the next section including the learning and the classification phase. Tracking is described in Section IV. Section V contains quantitative comparisons and experimental results and Section VI concludes the paper.

II. The bottom-up approach to 3D people detection

In this section we briefly summarize the bottom-up detector from [1]. Let us first define the terms. By *bottom-up* we

978-1-61284-380-3/11/$26.00 ©2011 IEEE

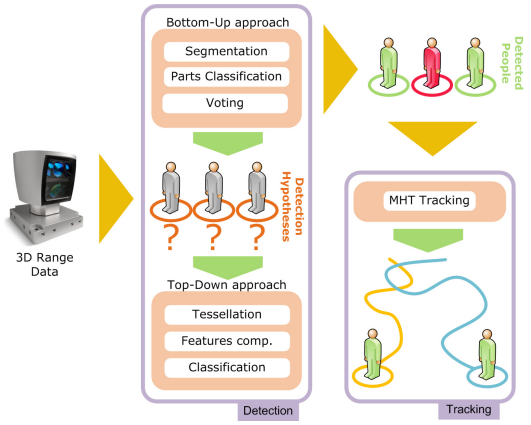

Fig. 2. Overview of the approach. The 3D range data are first processed by the bottom-up detector that generates candidate detection hypotheses (gray). They are verified by the top-down procedure. Confirmed detections (green) are finally tracked using a multi-hypothesis tracker (MHT).

denote the myopic scheme thats starts with local segments of 2D range data and works up to a detection hypothesis in 3D. By *top-down* we mean the procedure that begins with a predefined volume in 3D and the totality of data within that volume and works down to a classification of these data.

The idea of the method presented in [1] follows the observation that most 3D range finding devices acquire 3D point clouds in slices or lines of (typically non-coplanar) 2D data. They include, for instance, the Velodyne HDL-64E used here, the Alaska XT, or 2D laser scanners on turntables. This enables us, in a first step, to reuse known and proven techniques for detecting people in 2D range data that then are combined to a 3D detector.

A. The learning phase

As a first pre-processing step, each scan line is divided into segments using Jump Distance Clustering (JDC). JDC initializes a new segment each time the distance between two consecutive points exceeds a threshold θ_d. As a result, the data is reduced to a smaller number of segments. Then, each 2D segment is characterized by a set of geometrical and statistical features. A total of 17 features are taken for this task. They have been described in [8], [14], and [1] and include, for instance, measures of compactness, convexity, curvature, circularity, smoothness, etc.

To cope with the high level of variability in human articulation and shape, we take a part-based approach and subdivide humans into K different height layers. For each height layer, called part π^k, we create a classifier. This leads to a bank of K independent part classifiers, each one specialized for a certain height level. Segments of all scan lines that fall into a part contribute to the learning of the corresponding classifier. We use AdaBoost [15] for this task, a well known machine learning technique that has been proven successful already for detecting people in 2D range data [8].

To combine the individual classifications from the parts we use a voting model, loosely inspired by the implicit shape model approach [16]. To learn this voting model, for each segment $\mathcal{S}_i$ that belongs to a part π^k in the training set, we store the 3D displacement vector from the segment center to the center of the person. Then all displacement vectors for part π^k are collected and clustered using agglomerative clustering with average linkage. The resulting vector is called a vote. A weight is assigned to each clustered vote (see [1] for details) such that parts with a higher articulation variability (e.g. feet) receive a lower voting confidence.

Finally, as a practical interpretation of people in 3D we compute the average bounding box $\bar{\mathcal{B}} = \left(\bar{w}, \bar{d}, \bar{h}\right)$ from all person samples, where $\bar{w}, \bar{d}, \bar{h}$ represent the average width, depth and height, respectively. For the top-down detector, we align this volume by rotating it by the azimuth angle β of the line through the sensor origin and the center of the hypothesis (see Fig.3, right).

B. The detection phase

In the evaluation phase, we process newly arriving scans by the JDC segmentation step to obtain a set of segments. For each segment in each laser line, we compute the set of features and classify it using all K boosted part classifiers. We then compute a likelihood of the segment to belong to each part derived from the weighted linear combination of the weak classifiers. This likelihood serves as the weight of the vote that each part classifier casts.

People detection hypotheses are finally obtained as high density loci in the 3D voting space that are formed when multiple parts vote consistently. These loci are found using mean-shift mode estimation [17]. The score of a detection hypothesis is computed as a function of the vote weights, the number of votes, and the number of parts that contribute to the locus. For more details, please refer to [1].

III. A top-down approach to 3D people detection

With the approach just described, we achieved robust detection of people up to 10 m from the sensor with an EER of at least 95%. However, the EER dropped to 63% for a range up to 20 m since at this distance, people get sparsely sampled and are hard to distinguish from clutter. Let us consider the reasons.

Bottom-up detection hypotheses represent evidence that several segments at different heights vote consistently for being a part of a person. This method has the advantage of detecting people also from partial occlusions, when only a part of the body is visible. The drawback is however, that hypotheses can also be generated by segments in clutter that happen to cast votes in a consistent manner. Further, in certain situations, the approach can produce mode ambiguities in the voting space from symmetries. When two pedestrians walk side-by-side for instance, each person creates a strong mode in his or her center but also in between the two people originating from the arms and legs of both. Interestingly, these issues have also been noticed in visual voting-based object detection techniques [18].

To overcome these problems, we propose an additional top-down detection step that is still able to work with partial

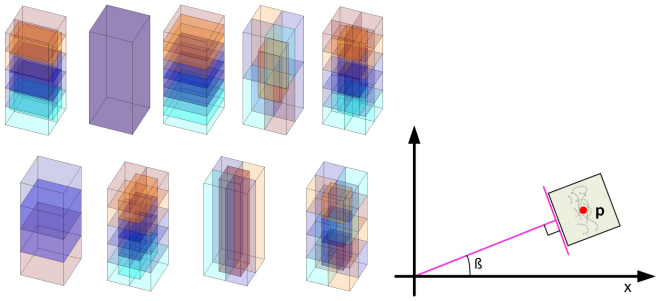

Fig. 3. **Left**: Tessellations generated to select the best volume subdivision for the top-down classifier. **Right**: Alignment of the volume for evaluating hypothesis **p** (top view).

body views. This classifier will treat the bottom-up detections as candidate hypotheses to be validated using an extended 3D person model. See Fig. 2 for an overview of the combined method.

A. The learning phase

Learning the top-down detector consists in two steps: learning the best volume tessellation and learning the features computed in this tessellation.

Volume tessellation

We aim to characterize the 3D shape of people by computing features in a tessellated volume $\bar{\mathcal{B}}$. We choose this subdivision approach since, in this way, local shape properties can be well described. This leads to the question of how a volume can be tessellated into sets of smaller volumes, a problem well known as tiling in computational geometry. For the sake of simplicity, we consider only axis-parallel (but non-cubic) voxels which reduces the complexity of the problem significantly. As there are still infinite numbers of tessellations of $\bar{\mathcal{B}}$, we define some constraints: we predefine the aspect ratios $\mathcal{V}$ that will be considered (to avoid voxels with extreme proportions), and define the list of increments $\Delta\mathbf{v}$ by which voxels will be enlarged. Each element $\mathbf{v}$ of $\mathcal{V}$ is a width-depth-height triplet (w, d, h) that is a multiplier coefficient of one of the voxel dimension.

The resulting procedure is Algorithm 1. The algorithm generates all possible voxel sizes (subject to $\mathcal{V}$ and $\Delta\mathbf{v}$), tests whether they can fill a volume $\mathcal{B}$ without gaps and subdivides $\mathcal{B}$ into a regular grid. The function $\texttt{Tess}(\mathcal{B}, w, d, h, \Delta_w, \Delta_d, \Delta_h)$ generates a regular face-to-face tessellation of $\mathcal{B}$ using the voxel (w, d, h), discarding protruding voxels and offsetting the resulting grid by $(\Delta_w, \Delta_d, \Delta_h)$. Calling this function with an offset produces voxels that also overlap each other (see also Fig.3, left). The algorithm generates gapless subdivisions of $\mathcal{B}$ that are complete in that no tessellation is missing under the given constraints.

We choose this tessellation approach as we want to be able to describe shape at different scales. Further, we want to allow the possibility that voxels can reflect the human anatomy, that is, can correspond to body parts such as legs or the head. Volume subdivisions with fixed-size voxels that do not overlap are less likely to establish such mappings.

Algorithm 1: Compute all axis-parallel tessellations $\mathcal{T}$ of a volume $\bar{\mathcal{B}}$.

Input: Volume $\mathcal{B}$, set of allowed voxel aspect ratios $\mathcal{V}$, list of voxel size increments $\Delta\mathbf{v}$.
Output: Set of all possible tessellations $\mathcal{T}$
$\mathcal{T} \leftarrow \{\}$
foreach $\Delta v_j \in \Delta\mathbf{v}$ **do**
  foreach $\mathbf{v}_k \in \mathcal{V}$ **do**
    $w = \Delta v_j \cdot \mathbf{v}_k^w$; $d = \Delta v_j \cdot \mathbf{v}_k^d$; $h = \Delta v_j \cdot \mathbf{v}_k^h$
    if $floor(w, w_{\mathcal{B}}) = 0 \wedge floor(d, d_{\mathcal{B}}) = 0) \wedge floor(h, h_{\mathcal{B}}) = 0$ **then**
      $\mathcal{T} = \mathcal{T} \cup \texttt{Tess}(\mathcal{B}, w, d, h, 0, 0, 0)$
      $\mathcal{T} = \mathcal{T} \cup \texttt{Tess}(\mathcal{B}, w, d, h, \frac{w}{2}, \frac{d}{2}, \frac{h}{2})$
    end
  end
end
return $\mathcal{T}$

3D features

Given a tessellation $\mathcal{T}_j$ we characterize shape by computing a set of geometrical and statistical features on the points within each voxel. Together, the feature values from all voxels make up an overall description of the human shape. Let the ith voxel of $\mathcal{T}_j$ be $\mathcal{T}_j^i$, then a feature f_k is formally defined as the mapping f_k: $\mathcal{T}_j^i \rightarrow \mathbb{R}$ that takes the N points $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ contained in $\mathcal{T}_j^i$ and returns a real value. We use the following nine features:

- *Number of points*: cardinality of $\mathcal{T}_j^i$ denoted as n. $f_1(\mathcal{T}_j^i) = n$
- *Sphericity*: this feature captures the level of sphericity from the ratio of the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ extracted from the scatter matrix computed on $\mathcal{P}$. $f_2(\mathcal{T}_j^i) = 3\frac{\lambda_3}{\sum_i \lambda_i}$ where $\lambda_1 > \lambda_2 > \lambda_3$.
- *Flatness*: this feature captures the degree of planarity from the eigenvalues. $f_3(\mathcal{T}_j^i) = 2\frac{\lambda_2 - \lambda_3}{\sum_i \lambda_i}$.
- *Linearity*: this feature captures the level of linearity from the eigenvalues. $f_4(\mathcal{T}_j^i) = \frac{\lambda_1 - \lambda_2}{\sum_i \lambda_i}$.
- *Standard deviation w.r.t. centroid*: $f_5 = \sqrt{\frac{1}{n-1}\sum_i (\mathbf{x}_i - \bar{\mathbf{x}})^2}$ where $\bar{\mathbf{x}}$ is the centroid. This is a measure of compactness.
- *Kurtosis w.r.t. centroid*: this feature is the fourth centralized moment of the data distribution in $\mathcal{T}_j^i$. $f_6(\mathcal{T}_j^i) = \sum_i (\mathbf{x}_i - \bar{\mathbf{x}})^4 / f_5(\mathcal{T}_j^i)$. This is a measure of peakedness.
- *Average deviation from median*: $f_7 = \frac{1}{n}\sum_i \|\mathbf{x}_i - \tilde{\mathbf{x}}\|$ where $\tilde{\mathbf{x}}$ is the vector of independent medians $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z})$. This is an alternative measure of compactness.
- *Normalized residual planarity*: sum of the squared error of a plane fitted into the data and normalized by n. $f_8 = \sum_i^n (a\,x_i + b\,y_i + c\,z_i + d)^2$ where a, b, c, d are the parameters of the plane derived from the eigenvalues of the scatter matrix. This is also a measure of flatness.
- *Number of points ratio*: this feature measures the point density of the voxel with respect to the entire volume $\mathcal{B}$: $f_9(\mathcal{T}_j^i) = \frac{n}{\|\mathcal{B}\|}$.

Training

The task for learning the top-down detector is to find the tessellation that best suits the shape characteristics of

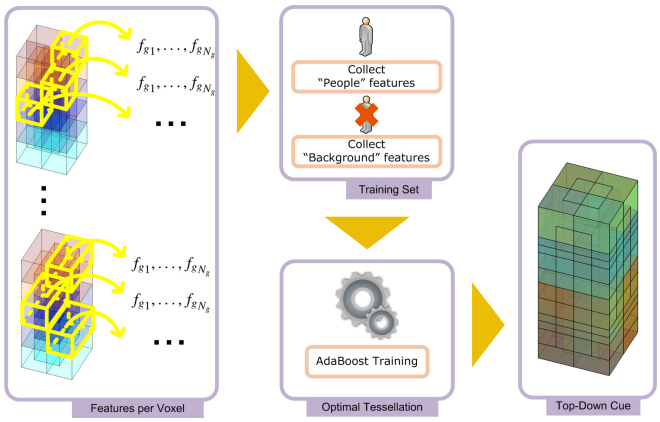

Fig. 4. Selecting the best volume tessellation and features. A set of features is computed on the points within each voxel. All features of all voxels over all tessellations are concatenated in a vector. These features are computed for all people and background samples and they constitute the training set for an AdaBoost classifier. The resulting strong classifier achieves the double objective of selecting the best features and the optimal tessellation. The final tessellation is shown and largely reflects the anatomy of the human articulated body.

humans in 3D point clouds and to select the most informative features for this purpose. The key idea is to *boost* both the tessellations and features (Fig. 4).

In a first step, we generate the training data by aligning the bounding box $\bar{\mathcal{B}}$ to the center of each positive sample (people) and negative sample (background). The outcome of this procedure is similar to a multiple view data set for people detection in images [19]. For each positive and negative sample, all features are computed for all voxels in the set of all tessellations $\mathcal{T}$. This leads to training vectors of the form $\mathbf{s} = (f_1^1, \ldots, f_9^1, f_1^2, \ldots, f_9^2, \ldots, f_1^V, \ldots, f_9^V)$ where V is the total number of voxels in the set of all tessellations $\mathcal{T}$. The training vectors along with their manually added label $\{-1, +1\}$ are then used to create an AdaBoost classifier with decision stumps as weak learners. The training set contains more negative than positive samples also to reflect the large variety of shapes that background point clouds can have. This imparity in the number of positive and negative samples requires the adaptation of the initial sample weights for which for follow the scheme in [1].

We train AdaBoost by adding weak classifiers until the misclassification error is smaller than 1%. The resulting strong classifier achieves a double objective, it selects the best possible features for classification (where 'best' is quantified by the voting weights), and implicitly selects the optimal subdivision $\mathcal{T}_{opt}$ of $\bar{\mathcal{B}}$. 'Implicitly' since $\mathcal{T}_{opt}$ is defined by the set of voxels in which the selected features have been computed. The method has the flexibility to select an arbitrary number of features in each voxel. A large number means that the voxel contains a particularly salient local shape. Note that the optimal tessellation can also be composed of a mixture of voxels from *different* tessellations.

It is interesting to interpret the learned optimal tessellation in Fig. 4. The subdivision largely reflects the anatomy of the human articulated body. The most stable parts of the body are the shoulder and above-knee areas which receive the highest resolution. The parts corresponding to feet, hands, arms, and the head (whereas the variability in the head area is caused by the different heights of people) receive lower resolutions. Further, the embedded vertically elongated voxels help to discriminate people on a rougher scale, characterizing the overall shape of humans as slender cylinders.

B. The combined detection approach

We finally put both detection cues together to form the proposed bottom-up top-down approach (Fig 2). As soon as a new scan arrives, the detection pipeline starts with the bottom-up detector that postulates a set of candidate detection hypotheses $\hat{\mathcal{Z}}$. Then, the volume $\bar{\mathcal{B}}$ is centered and aligned in these detection loci. The top-down detector comes into play by computing the selected feature set on all points in all voxels of the learned optimal tessellation $\mathcal{T}_{opt}$. Using this ensemble of feature values, the learned strong classifier validates the candidate hypotheses. Candidates for which the top-down classifier returns a positive response make up the set of positive detections $\mathcal{Z}(t)$ at time t.

IV. Multi Hypothesis Tracking

The detector presented so far is a single-frame detector that generates detection hypothesis from a single scan. To integrate this information over time, we track these hypotheses to estimate the 3D position and motion state of people, thereby smoothing the detector output.

For tracking, we employ a 3D implementation of the Multi-Hypothesis Tracking approach (MHT) by Reid [2], Cox *et al.* [3], and Arras *et al.* [20] able to filter full 3D states and measurements. It is beyond the scope of this paper to present this tracking approach in detail. We therefore give a brief summary.

The MHT belongs to the most general data association schemes that produces joint compatible assignments, integrates them over time, and is able to deal with the full life cycle of tracks (creation, confirmation, occlusion, and deletion) in a consistent probabilistic framework. The algorithm hypothesizes about all statistically feasible assignments between measurements and tracks and all possible interpretations of measurements as false alarms or new track and tracks as matched, occluded or obsolete.

At time t, each possible set of assignments and interpretations forms a data association hypothesis Ω_i^t (not to be confused with a detection hypothesis). Given the combinatorial complexity of assigning measurements to tracks and distributing interpretation labels, there are many possible hypotheses at each time step. Integrated over time, they form an exponentially growing tree that needs to be pruned. To this end, a probability is computed for each hypothesis, reflecting, for instance, the probability of measurements being false alarms or tracks entering the field of view. Then, only the best N_{hyp} hypotheses are retained. One of the key factors that makes the MHT a real-time procedure is that these N_{hyp} hypotheses can be determined directly in polynomial time.

The tracker produces a set of tracks that estimate the filtered position and velocity of people in 3D (see Fig. 1 for an example track).

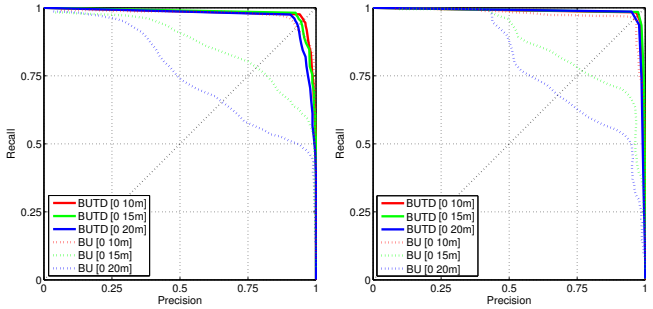

Fig. 5. Comparison of the bottom-up top-down detector (BUTD) with the bottom-up (BU) approach in [1]. The Equal Error Rate (EER) of the BUTD approach is at least 93% over all ranges in both data sets. The performance gain of BUTD over BU is particularly evident for far away detections. **Left**: Precision recall graph for the Tannenstrasse data set. **Right**: Precision recall graph for the Polyterrasse data set.

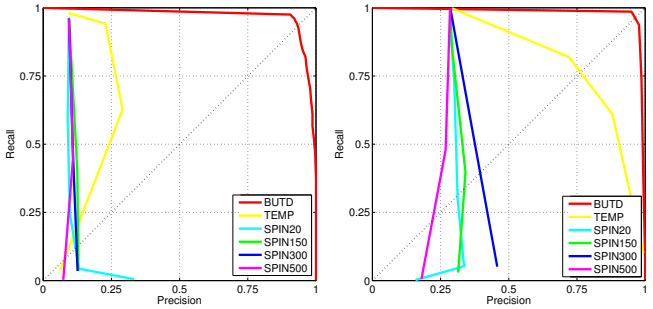

Fig. 6. Comparison of the bottom-up top-down detector (BUTD) with the template based approach (TEMP) and the spin image-based bag-of-words approach with several dictionaries (SPIN20-SPIN500) in $[0, 20m]$ range. None of the compared technique achieves the performance of BUTD. The SPIN method is apparently not able to generalize the articulated shape of people and/or their sparse representation at large distances. Further, a large dictionary (500 words) leads to overfitting. TEMP achieves good results only in the data set with simple background. **Left**: Precision recall graph for the Tannenstrasse data set. **Right**: Precision recall graph for the Polyterrasse data set.

V. Experiments

We evaluate our algorithm on two outdoor data sets collected with a Velodyne HDL 64E S2 laser scanner. The first data set, named Polyterrasse, has been collected in a large area in the front of the ETH Zurich main building, accessible only to people and bicycles. The second data set, Tannenstrasse, has been collected on a busy street crossing in downtown Zurich with trams, cars, pedestrians, and bicycles. We have collected 900 full-view point clouds for the first data set and 500 for the second data set. The sensor rotates with a frequency of 5Hz at a maximum range limited to $20m$. This produces around 120,000 points per 3D scan.

A. Training

We labeled each frame manually by placing bounding boxes around people if they are represented by at least 200 points and exceed $1.20m$ in height. A second type of annotation, explained later, is made for people with at least 100 points and $1m$ height. We considered only subjects with heights $\pm 15cm$ from the mean. This leads to 455 samples of people from all possible view points, standing still and walking, and 6025 samples of background. The constraints for the tessellation algorithm are $\Delta\mathbf{v} = (0.2m, ..., 0.8m)$ and $\mathcal{V}$ being $\{1, 1, 2.5\}, \{1, 1, 5\}, \{1, 1, 1.25\}, \{1, 1, 1\}, ..., \{4, 4, 4\}$, and the permutations of $\{1, 1, 2\}, \{1, 1, 3\}, \{2, 2, 3\}, \{4, 4, 3\}, \{4, 4, 2\}$ These constraint produce a minimal voxel dimension of $0.2m$. This rather large value has been set to ease detection in case of imprecise placement of detection hypotheses with respect to the object centers. The training error objective for the top-down approach of 1% has been reached with 95 decision stumps.

B. Detector Evaluation

In this section we present experimental results and comparisons with other detection techniques. For the quantitative analysis, we adopt the no-reward-no-penalization policy from [21]. The policy does not count true or false positives when a detection matches an annotation of the second type.

In the first experiment, we compare the performance of the combined bottom-up top-down detector (BUTD) with the bottom-up approach presented in [1] (Fig 5). The Equal Error Rates (EER, where precision equals recall), of the new approach over the distance ranges from the sensor of $[0, 10m]$, $[0, 15m]$, and $[0, 20m]$ are 95.4%, 94.5%, and 93.4% for the Tannenstrasse data set and 97.6%, 97.2%, and 96.2% for the Polyterrasse data set, respectively. BUTD clearly achieves much better performance in terms of precision-recall, particularly for individuals at large distances from the sensor where BUTD shows an improvement of over 30% over the BU technique. Thus, the main role of the top-down detector is to remove false positives and to reinforce weak detections of the BU approach. Similar insights have been gained for other bottom-up top-down detectors in computer vision [16]. We see the identical improvement in both data sets as indication for good generalization capabilities of the new algorithm.

In a second set of experiments, we compare the proposed approach to two established 3D object detection techniques: spin images (SPIN) and template-based classification (TEMP). We first describe our implementation of both techniques before discussing the results.

TEMP training: To each positive training sample, we align the volume $\bar{\mathcal{B}}$ as shown in Fig.3 and subdivide it using cubic cells of size $0.2m$. The cells represent bins of a template histogram $\mathcal{H}$. If a point of a sample lies within a cell, the associated histogram bin is incremented. All positive training samples are used to learn the model $\mathcal{H}_m$. The histogram is then normalized to one.

TEMP detection: To every candidate detection hypothesis from the BU technique, we align the volume $\bar{\mathcal{B}}$ and compute a histogram $\mathcal{H}_z$ using the same procedure employed for training. The histograms are then matched using an Euclidean distance $d(\mathcal{H}_m, \mathcal{H}_z)$.

The SPIN approach follows the 3D bag-of-words detection scheme [22]. The method represents point clouds as collections of local descriptors ignoring their spatial disposition.

SPIN training: We compute spin image descriptors for both the positive sample set and the negative set. For each

sample we randomly draw 160 spin image descriptors in a regular 3D grid of $0.2m$ resolution in the volume $\bar{\mathcal{B}}$. Spin images have a resolution of $0.1m$ in each dimension. $289,415$ descriptors are collected and clustered using k-means with an implementation optimized for large data sets [23]. The number of clusters is varied to be $20, 150, 300$ and 500 which creates several dictionaries of words where each word corresponds to a cluster center. From these words a histogram is derived that holds the frequency of how often a word appears in the considered sample. Finally, we create an AdaBoost classifier from the set of all positive and all negative histograms, thereby boosting the words, using 100 decision stumps.

SPIN detection: The same grid used for training is used for sampling spin images around all candidate detection hypothesis. Each spin image is matched to every word of a dictionary and associated to the one with the smallest Euclidean distance. These words are then used to build the histogram that is fed into the learned strong classifier.

The results for the TEMP comparison for the range $[0, 20m]$ are shown in Fig. 6. The method clearly underperforms with respect to the new BUTD method on both data sets. It achieves an EER of more than 75% on the Polyterrasse data sets but mainly due to the simplicity of the background. For the more complex Tannenstrasse data set, the EER is as low as 7.7% over the $[0, 20m]$ range.

The results for the SPIN comparison for the range $[0, 20m]$ are shown in Fig 6. The EER for each dictionary is 11.8%, 12.7%, 12.7%, 10.8% for the Tannenstrasse and 31.2%, 33.6%, 39.5%, 21.9% for the Polyterrasse data sets. For none of its parameters, the SPIN approach is able to be on par with our detector. It is interesting to note that a large dictionary (500 words) causes overfitting and the classifier underperforms with respect to more compact dictionaries. Similar effects have been noticed in the field of computer vision [24].

Discussion: The poor performance of the template-based technique is apparently caused by the fact that it is not based on a discriminative classifier and that shape is merely encoded by distributions of point densities over a fixed tessellation. The spin image descriptors also rely on a fixed space subdivision and a single feature (projected distances onto a fitted local plane). In contrast, our method boosts a set of features (in which point density is just one) and *learns* the tessellation. Thus, we are able to learn the locations and scales at which features are computed to characterize shape stably and robustly, enabling the method to deal with complex articulated shapes. Another reason for the poor performance of the spin image-based technique is that descriptors depend on the estimation quality of local normal vectors. For sparse and noisy measurements, which is the case for this kind of 3D data, this can pose a problem. A major advantage of our top-down detector is that it has no parameters. There are constraints for the tessellation algorithm or a misclassification bound for AdaBoost. But this is unlike the spin image bin resolution, the scale (number of bins) or the number of dictionary words. These parameters strongly influence the behavior of the methods. To find good values typically requires tuning and experience.

TABLE I

TRACKING PERFORMANCE COMPARISON: BUTD VS BU.

Data set	MOTP	MOTA	FN	FP	ID
BU (Tann)	$< 0.16m$	26.2%	51.4%	22.1%	6
BUTD (Tann)	$< 0.16m$	73.6%	16.9%	8.2%	30
BU (Poly)	$< 0.16m$	23.1%	18.7%	57.7%	11
BUTD (Poly)	$< 0.16m$	89.1%	2.6%	7.6%	20

A C++ implementation of the bottom-up top-down detector, not optimized for speed, processes around 120,000 points per 3D scan and runs at $\sim 1Hz$ on a standard PC.

C. Tracking Evaluation

To evaluate our 3D tracking results, we use the CLEAR MOT metrics [25] as measure. The metric counts three numbers with respect to the ground truth that are incremented at each frame: misses (missing tracks that should exist at a ground truth position, FN), false positives (tracks that should not exist, FP), and mismatches (track identifier switches, ID). From these numbers, two values are determined: MOTP (average metric distance between estimated targets and ground truth) and MOTA (the average number of times of a correct tracking output with respect to the ground truth).

For the purpose of this evaluation, we manually annotated the tracks in both data sets based on both types of annotations. This includes the annotations used for the no-rewards-no-penalty policy that are particularly difficult to detect due to a low point density.

We compare the MHT tracking performance using the BU detector and the combined method (see Table I). The experiments have been run with $N_{Hyp} = 100$ number of hypotheses where larger numbers for N_{Hyp} have not led to a different behavior. The performance gain of the BUTD method versus the BU detector is evident for both data sets. The MOTA index is 2.8× higher in the Tannenstrasse and 3.8× higher in the Polyterrasse data set. The MOTP index is constant in all cases. In both data sets, the value of FN is at least 3× smaller while the value of FP is at least 2.8× smaller. The number of track confusions, ID, is higher for the combined approach simply because many more targets are tracked over longer periods of time. In comparison, the BU method is only able to track targets shortly and at close range to the sensor.

The average cycle time of our MHT system, optimized for speed, is ≤ 5ms for each 3D scan on a standard PC.

VI. CONCLUSIONS

In this paper we presented a combined bottom-up top-down detector for people in 3D range data. Unlike prior work, no ground plane assumption is needed. The bottom-up detector generates candidate detection hypotheses that are validated by a top-down procedure. The top-down method relies on a novel technique that boosts both features and the volume tessellation in which the features are computed. This

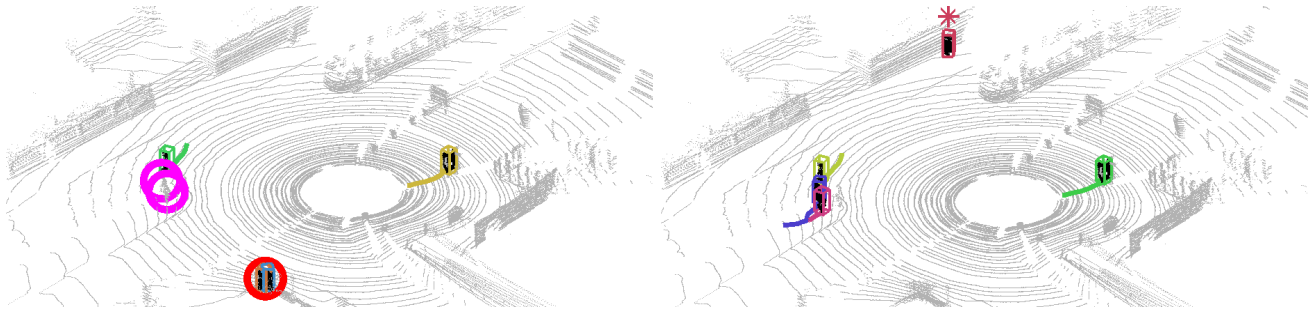

Fig. 7. Qualitiative comparison of MHT tracking and detection with BUTD and BU in an urban scene (Tannenstrasse data set). Red circles represent false positives, magenta circles false negatives. **Left** Bottom-up detection only: false positives are detected in clutter and generate false tracks, two people are not detected due to point cloud sparsity. **Right** Bottom-up top-down detection, every person has been correctly detected and tracked, the asterisk indicates correct detection of a sparsely described target (annotation of the second type).

allows the detector to characterize local 3D shape on multiple scales and at different locations, giving it the ability to stably deal with sparsely sampled and articulated objects.

In experiments with two different data sets in cluttered urban environments we achieved highly robust detection of people in up to $20m$ distance from the sensor with equal error rates of at least 93%. The method clearly outperforms two established 3D object recognition methods namely spin images and template-based classification. Opposed to these methods, the top-down detector is also parameter-free.

The confirmed detection hypotheses are then fed into a 3D implementation of a multi-hypothesis tracker. We quantified how much the newly introduced top-down detector also improves tracking in comparison with the bottom-up detector using the CLEAR MOT metrics. The experiments show that we are able to robustly track people in 3D.

Acknowledgements

The authors thank Rudolph Triebel for the fruitful discussions regarding classification of 3D data. This work has been supported by the German Research Foundation (DFG) under contract number SFB/TR-8.

References

[1] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart, "A layered approach to people detection in 3D range data." in *AAAI Conf. on Artif. Intell. (AAAI)*, 2010.

[2] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[3] I. J. Cox and S. L. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 18, no. 2, pp. 138–150, 1996.

[4] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Trans. on Pattern Analysis & Machine Intell.*, vol. 21, no. 1, pp. 433 – 449, May 1999.

[5] B. Kluge, C. Köhler, and E. Prassler, "Fast and robust tracking of multiple moving objects with a laser range finder," in *Int. Conf. on Rob. & Autom. (ICRA)*, 2001.

[6] A. Fod, A. Howard, and M. Mataríc, "Laser-based people tracking," in *Int. Conf. on Rob. & Autom. (ICRA)*, 2002.

[7] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "People tracking with a mobile robot using sample-based joint probabilistic data association filters," *Int. Journ. of Rob. Research*, vol. 22, no. 2, pp. 99–116, 2003.

[8] K. O. Arras, O. Martínez Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Int. Conf. on Rob. & Autom. (ICRA)*, 2007.

[9] S. Gidel, P. Checchin, C. Blanc, T. Chateau, and L. Trassoudaine, "Pedestrian detection method using a multilayer laserscanner: Application in urban environment," in *Int. Conf. on Intel. Rob. and Sys. (IROS)*, 2008.

[10] A. Carballo, A. Ohya, and S. Yuta, "Fusion of double layered multiple laser range finders for people detection from a mobile robot," in *IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2008.

[11] O. M. Mozos, R. Kurazume, and T. Hasegawa, "Multi-part people detection using 2d range data," *International Journal of Social Robotics*, vol. 2, no. 1, 2010.

[12] L. Navarro-Serment, C. Mertz, and M. Hebert, "Pedestrian detection and tracking using three-dimensional ladar data," in *Int. Conf. on Field and Service Robotics*, 2009.

[13] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. Matthies, "Results from a real-time stereo-based pedestrian detection system on a moving vehicle," in *Workshop on People Detection and Tracking, IEEE ICRA*, 2009.

[14] L. Spinello, R. Triebel, and R. Siegwart, "Multimodal people detection and tracking in crowded scenes," in *AAAI Conf. on Artif. Intell. (AAAI)*, 2008.

[15] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Jour. of Comp. and System Sciences*, vol. 55, no. 1, 1997.

[16] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 2005.

[17] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis & Machine Intell.*, vol. 24, pp. 603–619, 2002.

[18] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *In ECCV wshop on stat. learning in comp. vis.*, 2004, pp. 17–32.

[19] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 1997.

[20] K. O. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *Int. Conf. on Rob. & Autom. (ICRA)*, 2008.

[21] M. Enzweiler and D. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Trans. on Pattern Analysis & Machine Intell.*, vol. 31, no. 12, pp. 2179–2195, 2009.

[22] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Wshop on Stat. Learning in Computer Vision, ECCV*, 2004.

[23] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. on Pattern Analysis & Machine Intell.*, 2000.

[24] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Eur. Conf. on Comp. Vis. (ECCV)*, 2006.

[25] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.