Real-Time Full-Body Human Attribute Classification in RGB-D Using a Tessellation Boosting Approach

Timm Linder Kai O. Arras

***Abstract*— Robots that cooperate and interact with humans require the capacity to detect and track people, analyze their behavior and understand human social relations and rules. A key piece of information for such tasks are human attributes like gender, age, hair or clothing. In this paper, we address the problem of recognizing such attributes in RGB-D data from varying full-body views. To this end, we extend a recent tessellation boosting approach which learns the best selection, location and scale of a set of simple RGB-D features. The approach outperforms the original approach and a HOG baseline for five human attributes including *gender*, *has long hair*, *has long trousers*, *has long sleeves* and *has jacket*. Experiments on a multi-perspective RGB-D dataset with full-body views of over a hundred different persons show that the method is able to robustly recognize multiple attributes across different view directions and distances to the sensor with accuracies up to 90%. Our methods runs in real-time, achieving a classification rate of around 300 Hz for a single attribute.**

Fig. 1. Extracted RGB-D point clouds with corresponding classification results for three different human attributes. The text at the top of each image is the attribute in question and the number below each image the confidence of the boosted classifier, where the sign indicates an association with the negative or positive class.

I. Introduction

The ability to describe and reidentify individual persons that interact with a robot is an important perceptual skill for robots in human environments, for example, when providing personalized services. The task of extracting human attributes such as gender, age group, or clothing-related attributes from a person's appearance is relatively unexplored for RGB-D data although particularly relevant for robotics. Unlike methods that rely on image data only, which may strongly suffer from varying illumination conditions when deployed on a mobile robot, RGB-D data are typically less sensitive to indoor ambient conditions and provide 3D point clouds that allow for the extraction of geometric cues in addition to visual appearance. There is a limited number of works in this area, focussing e.g. on gender recognition [1], color-based clothing attributes [2], person re-identification [3], [4], or requiring accurate on-line estimates of the skeleton joint angles [5].

In this paper, we extend our previous work on full-body human gender recognition in RGB-D data [1] for the task of more generally recognizing human attributes. Concretely, we make the following contributions:

- We propose a novel and particularly efficient recognition approach for human attributes from RGB-D data. This is achieved by extending a tessellation boosting approach, originally developed for people detection in 3D data, with geometric extent features and RGB color features from multiple color spaces. While simple and fast to compute, the color features alone are able to clearly improve classification accuracy for several types of attributes.
- Pre-processing: We scale the input person cloud so as to ensure that it fills the bounding volume used during training and we apply a voxel filter on the generated tessellations before boosting which reduces the feature descriptor length significantly, typically by 80%. This results in improved classification accuracies and reduced memory consumption and classifier training times.
- We evaluate our approach on a large dataset with RGB-D point clouds of 118 persons in both standing and walking poses under various view angles and distances to the sensor. To this end, we extended the dataset – first presented in [1] – with annotations of multiple human attributes such as *has long trousers*, *has long sleeves*, *has long hair*.

The authors are with the Social Robotics Lab, Dept. of Computer Science, University of Freiburg, Germany. http://srl.informatik.uni-freiburg.de, {linder,arras}@cs.uni-freiburg.de. This work has been partly supported by the European Commission under contract number FP7-ICT-600877 (SPENCER).

II. Related work

Previous research on human attribute recognition mainly focusses on the detection of single attributes like gender [6]–[9], many of them use specialized methods applied to frontal face images. State-of-the-art methods achieve up to 90% accuracy for gender on large datasets with several thousands of training images.

A smaller group of works use full-body views which is challenging due to the high variability in human appearance, posture, and distance from the sensor. Out of these, only few exploit 3D or RGB-D data [1], [4], [10].

Recently, progress has been made in extracting more localized, appearance-based human attributes from image

data. Bourdev *et al.* [11] learn 1200 *poselets*, which represent small parts of the human body under a specific pose, in order to decompose view point and pose from appearance and recognize attributes which describe gender, hair style, and clothing of a person in color image data. The feature vectors computed on the poselets, which are then fed into three layers of SVM classifiers, incorporate histogram of oriented gradients (HOG), HSB color histogram and skin mask features. Example attributes include *is male*, *has hat*, *has t-shirt*, *has shorts*, *has glasses*, or *has long pants*. The authors report an average precision (AP) of around 82.4% for gender, 73% for long hair, 74% for long sleeves and 90% for long pants on a database of 8000 color images containing persons in a large number of different poses and viewpoints under mostly favorable illumination conditions.

In [12], the authors replace the SVM layers and the manually crafted HOG and histogram features by a convolutional neural network for each individual poselet as well as the overall image. The outputs of the networks are then combined using a linear classifier for binary attribute prediction. The decomposition into poselets helps to decouple the influences of pose and viewpoint on appearance from weaker but relevant signals of certain human attributes, such as *wears glasses*. The approach is able to increase the mean average precision by 13% over the results of [11]. However, they require large datasets: images of around 25,000 people are extraced from Facebook, in addition to 8000 color images from an existing database to avoid overfitting.

Using RGB-D data, the only approach to our knowledge is due to Wang *et al.* [5] who recognize multiple attributes using a similar approach to [11], [12] in that individual SVM classifiers are trained on body parts instead of the whole body. Instead of poselets, they use skeleton-based full-body pose estimates provided by the Kinect sensor to generate sampling regions around body limbs for the computation of HOG, LBP, Gabor filter and color+depth histograms. Trained on a dataset with full-body views of over 4,600 persons recorded by a sensor mounted above the entrance of an elevator, they outperform a reimplementation of [11] and achieve equal precision and recall (EPR) rates of 87% on gender, 87.4% on short sleeves and 91.6% on long pants.

Unlike these methods, our approach does not require skeleton estimates – which can be hard to come by under wider ranges of condition – and fully leverages 3D shape data that is relatively robust with regard to illumination conditions. As a result, our method achieves already good recognition accuracies on 3D points cloud data only. Finally, due to its simplicity, our approach is extremely efficient at 125 Hz on a single CPU core (300 Hz on four cores) without GPU acceleration, which is highly relevant for real-time implementations on resource-constrained mobile robots.

III. Our method

In this section, we first describe our tessellation boosting approach originally developed for people detection in 3D data [13] and applied to the task of full-body human gender recognition [1]. We then describe our extensions, the new geometric extent and color features and two pre-processing steps.

Our tessellation-based method for 3D object characterization was, in its original form, first used in [13] for the task of person detection in 3D range data. The method takes a bottom-up top-down approach where we classify object detection hypotheses from a bottom-up classifier using a learned top-down model. The bottom-up classifier can either be a simple region-of-interest (ROI) detector or a more sophisticated detector, typically tuned for higher recall. Here, we focus on the top-down method and assume to have a simple ROI detector which extracts candidate person detections from the scene in the form of RGB-D point clouds, like the examples shown in Fig. 1.

The top-down method characterizes the point cloud by a set of features computed on the measurements within axis-aligned voxels of the 3D object and uses AdaBoost to create a strong classifier with the best features and voxels. What distinguishes this method is that the boosted classifier not only selects the best features and thresholds, but also the best combination of voxels on which these features have found to be informative. Thus, the classifier jointly learns the best scales and locations of features on the 3D object for the classification task at hand. This allows to robustly and stably describe complex articulated shapes, as shown in [1] for the example of gender recognition, where the learned tessellation outperformed a fixed tessellation by a large margin of 6-10%.

A. Tessellation Generation

We assume persons to fit into a fixed-size bounding volume $\mathcal{B}$, centered around the median in x and y of the point cloud. The size of $\mathcal{B}$ can either be fixed and taken from the maximum expected object size or learned from a training set as in [13].

We subdivide the volume into voxels which leads to the question of how a volume can be tessellated into a collection of smaller volumes, a problem well known as tiling in computational geometry. For the sake of simplicity, we consider only axis-parallel voxels which reduces the complexity of the problem but still leaves an infinite number of tessellations of $\mathcal{B}$. Thus, we define a set of proportion constraints $\mathcal{C}$ to exclude extreme aspect ratios of voxels and a list of increments $\mathbf{s}$ by which voxels will be enlarged. Each element $\mathbf{c} = (w, d, h) \in \mathcal{C}$ is a width-depth-height triplet with multipliers of the respective voxel dimension.

The resulting procedure, Algorithm 1, generates all possible voxel sizes subject to $\mathcal{C}$ and $\mathbf{s}$. Defining the remainder after ceiling-division $\texttt{rem}(a, b)$ as $|a - \lceil \frac{a}{b} \rceil b|$, the algorithm tests whether voxels can fill a volume $\mathcal{B}$ without gaps and subdivides $\mathcal{B}$ into a regular grid. The function $\texttt{Tess}(\mathcal{B}, w, d, h, \Delta_w, \Delta_d, \Delta_h)$ produces a regular face-to-face tessellation of $\mathcal{B}$ with voxels of size (w, d, h) and offset $(\Delta_w, \Delta_d, \Delta_h)$ to also allow voxels that overlap each other. The algorithm generates gapless subdivisions of $\mathcal{B}$ that are complete in that no tessellation is missing under the given constraints. We also allow slightly protruding voxels with a tolerance θ.

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

Input: Bounding volume $\mathcal{B}$ of size $w_\mathcal{B} \times d_\mathcal{B} \times h_\mathcal{B}$, set of voxel proportion constraints $\mathcal{C}$, list of voxel scaling factors $\mathbf{s}$, protrusion tolerance θ. **Output:** Set of all possible tessellations $\mathcal{T}$

$\mathcal{T} \leftarrow \{\}$
foreach $s_j \in \mathbf{s}$ **do**
  foreach $\mathbf{c}_k = (w_k, d_k, h_k) \in \mathcal{C}$ **do**
    $w = s_j \cdot w_k;\ d = s_j \cdot d_k;\ h = s_j \cdot h_k$
    if $\texttt{rem}(w_\mathcal{B}, w) < \theta \wedge \texttt{rem}(d_\mathcal{B}, d) < \theta \wedge \texttt{rem}(h_\mathcal{B}, h) < \theta$ **then**
      $\mathcal{T} \leftarrow \mathcal{T} \cup \texttt{Tess}(\mathcal{B}, w, d, h, 0, 0, 0)$
      $\mathcal{T} \leftarrow \mathcal{T} \cup \texttt{Tess}(\mathcal{B}, w, d, h, \frac{w}{2}, \frac{d}{2}, \frac{h}{2})$
    end
  end
end
return $\mathcal{T}$

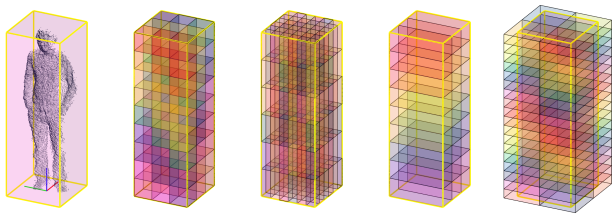

Fig. 2. Left: person candidate point cloud, centered around the median in x and y. The other pictures show example tessellations of the bounding volume $\mathcal{B}$ generated using our tessellation algorithm. We also allow protruding voxels, shown in the rightmost picture.

As constraints we use scaling factors $\mathbf{s} = (0.1, 0.2, ..., 0.8)\ [m]$ and proportions $\mathcal{C}$ being the set of all permutations of $\{\{1,1,1\}, \{1,1,1.25\}, \{1,1,2\}, \{1,1,2.5\}, \{1,1,3\}, \{1,1,4\}, \{1,1,5\}, \{1,1,6\}, \{1,1,8\}, \{1,1,10\}, \{2,2,3\}, \{4,4,2\}, \{4,4,3\}\}$. These lead to 134 valid tessellations, of which some examples are shown in Fig. 2.

B. Classifier Training

Let $\mathcal{T}_j$ be the jth valid tessellation and $\mathcal{V}_j^i$ its ith voxel. Then, for each $\mathcal{V}_j^i$ of all generated $\mathcal{T}_j$'s, we determine the set $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of points inside the voxel's volume. With the goal to describe shape properties locally, we then compute a set of RGB-D point cloud features f_i that characterize geometrical and statistical properties of $\mathcal{P}$, see Table I. Most of them can be computed very efficiently from the points' scatter matrix via eigenvalue decomposition and none of them require estimation of the surface normals.

Training samples are formed by stacking the features of all voxel point clouds of all tessellations into one large feature vector and associating the corresponding ground truth class label. We train an AdaBoost classifier with n_{weak} decision stumps as weak learners. After training, the final model is given by the collection of all voxels in which *at least one feature* has been selected. The resulting strong classifier achieves a double objective, it selects the best features ('best' quantified by the AdaBoost voting weights) and selects the optimal subdivision $\mathcal{T}_{opt}$ of $\mathcal{B}$ for the classification task at hand. The method can select an arbitrary number of features in each voxel – a large number, for instance, means that the voxel contains a particularly salient local shape – and may also select a mixture of voxels from *different* tessellations. This implicit feature selection is performed separately for each human attribute, yielding an attribute-specific classifier.

#	Description	Expression
1	Number of points	The point count of $\mathcal{P}$ denoted as n. $f_1 = n$
2	Density	Captures the normalized point density w.r.t. the entire point cloud: $f_2 = \frac{n}{N_\mathcal{B}}$
3	Sphericity	Captures the level of sphericity from the ratio of the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ extracted from the scatter matrix of $\mathcal{P}$. $f_3 = 3\frac{\lambda_3}{\sum_i \lambda_i}$ where $\lambda_1 > \lambda_2 > \lambda_3$
4	Flatness	Measures the degree of planarity from the eigenvalues. $f_4 = 2\frac{\lambda_2 - \lambda_3}{\sum_i \lambda_i}$
5	Linearity	Captures the level of linearity from the eigenvalues. $f_5 = \frac{\lambda_1 - \lambda_2}{\sum_i \lambda_i}$
6	Standard deviation w.r.t. centroid	Measures the compactness of points in $\mathcal{P}$, $f_6 = \sqrt{\frac{1}{n-1}\sum_i (\mathbf{x}_i - \bar{\mathbf{x}})^2}$ where $\bar{\mathbf{x}}$ is the centroid.
7	Kurtosis w.r.t. centroid	Captures the peakedness of points in $\mathcal{P}$, fourth centralized moment of the data distribution in $\mathcal{P}$. $f_7 = \sum_i (\mathbf{x}_i - \bar{\mathbf{x}})^4 / f_6$.
8	Average deviation from median	Alternative measure of compactness. $f_8 = \frac{1}{n}\sum_i \|\mathbf{x}_i - \tilde{\mathbf{x}}\|$ where $\tilde{\mathbf{x}}$ is the vector of independent medians $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z})$.
9	Normalized residual planarity	Alternative measure of flatness. Squared error sum of a plane fitted into $\mathcal{P}$ normalized by n. $f_9 = \sum_i^n (a\,x_i + b\,y_i + c\,z_i + d)^2$ where a, b, c, d are the parameters of the plane derived from the eigenvalues of the scatter matrix.

TABLE I

GEOMETRIC FEATURES FROM [1], [13]

#	Description	Expression
10	Depth	Geometric extent of $\mathcal{P}$ in x direction. $f_{10} = \max_\mathcal{P}(x_i) - \min_\mathcal{P}(x_i)$
11	Width	Geometric extent of $\mathcal{P}$ in y direction. $f_{11} = \max_\mathcal{P}(y_i) - \min_\mathcal{P}(y_i)$
12	Height	Geometric extent of $\mathcal{P}$ in z direction. $f_{12} = \max_\mathcal{P}(z_i) - \min_\mathcal{P}(z_i)$
13 – 15	RGB component-wise mean	Mean of the red, green, blue components of each point in $\mathcal{P}$. $(f_{13}, f_{14}, f_{15}) = (\bar{r}, \bar{g}, \bar{b})$
16 – 18	RGB component-wise standard deviation	Standard deviation of the red, green, blue components of each point in $\mathcal{P}$. $(f_{16}, f_{17}, f_{18}) = (\sigma(r), \sigma(g), \sigma(b))$
19 – 21	HSV component-wise mean	Mean of the hue, saturation, value components of each point in $\mathcal{P}$. $(f_{19}, f_{20}, f_{21}) = (\bar{h}, \bar{s}, \bar{v})$
22 – 24	HSV component-wise standard deviation	Standard deviation of the hue, saturation, value components of each point in $\mathcal{P}$. $(f_{22}, f_{23}, f_{24}) = (\sigma(h), \sigma(s), \sigma(v))$
25 – 27	$\mathrm{Y'C_BC_R}$ component-wise mean	Mean of the luma, blue-difference, red-difference components of each point in $\mathcal{P}$. $(f_{25}, f_{26}, f_{27}) = (\bar{Y'}, \bar{C}_B, \bar{C}_R)$
28 – 30	$\mathrm{Y'C_BC_R}$ component-wise standard deviation	Standard deviation of the luma, blue-difference chroma, red-difference chroma components of each point in $\mathcal{P}$. $(f_{28}, f_{29}, f_{30}) = (\sigma(Y'), \sigma(C_B), \sigma(C_R))$

TABLE II

NEW GEOMETRIC EXTENT AND COLOR FEATURES

C. Geometric extent and color features

In the current set of geometric point cloud features listed in Table I, *standard deviation w.r.t. centroid* and *average deviation from median* measure the compactness of the points in $\mathcal{P}$. However, these are based upon point-to-point distances and do not capture well the geometric extents of $\mathcal{P}$, useful e. g. to distinguish vertically and horizontally elongated shapes that are approximately aligned to the axes. To achieve this, we extend the feature set by *depth*, *width* and *height* of $\mathcal{P}$ defined in Table II.

All features considered so far are geometric in nature and do not encode color information – clearly a very informative object property. We thus extend the feature set by several scalar RGB features. Concretely, we compute the average color within the points $\mathcal{P}$ of a volume $\mathcal{V}_j^i$ by computing the component-wise mean of the red, green and blue channels as well as their standard deviation. Additionally, since RGB color values are not invariant with regard to illumination, we also compute mean and standard deviation in the HSV color space, were H stands for hue, S for saturation and V for value (brightness). The expectation here is that the V feature should be chosen less often by the AdaBoost classifier due to its dependence on lighting. Finally, we also conduct experiments in the $\mathrm{Y'C_BC_R}$ color space, where $\mathrm{C_B}$ and $\mathrm{C_R}$ represent the blue-difference and red-difference chroma components, which are illumination-independent, and $\mathrm{Y'}$ the luma. The advantage of this color space, used e.g. in JPEG compression and digital video, is that skin colors are very localized in the chroma channels and that their values do not vary much between subjects of different ethnicity, which can be useful for robustly localizing skin colors (i. e. uncovered body parts) in the point cloud. As for the other color spaces, we compute the component-wise mean and standard deviation within a given set of points $\mathcal{P}$.

The resulting list of new features is shown in Table II.

D. Scaling of input clouds and voxel filter

So far, we have assumed a fixed-size bounding box $\mathcal{B}$ in which voxels are generated. The size of this bounding volume can be learned from training data or set to some upper bound. This, however, is problematic when there is a lot of variation in person size: if the training set includes very large and very small subjects (e. g. children), the classifier may fail to locate specific scales on the human body that are informative for a particular attribute (e. g. hips and waist for gender, or the head for *long hair*), or it might at least spend a significant number of weak classifiers to accommodate for the differences in size. Learning multiple classifiers for different person sizes, and then selecting the appropriate classifier from the point cloud beforehand, requires even more training data to cover different person sizes across all the attributes. As a more effective alternative to deal with this issue, we scale the input person cloud in z direction to a fixed size. In our case, we stretch the point cloud uniformly to a height of $h = 1.8\mathrm{m}$, leaving the x and y coordinates unaltered.

We also discard non-informative voxels in a filtering step, which reduces both memory consumption and classifier training times. The feature matrix F contains the stacked features of all samples over all voxels and tessellations. The entire matrix, whose size $|\mathcal{V}_j^i| \cdot n_{\mathrm{features}} \cdot n_{\mathrm{samples}}$ scales linearly with the number of voxels $|\mathcal{V}_j^i|$, requires, for instance, with just the nine geometric features from Table I, 12 GB RAM on the full dataset. Thus, for each voxel $\mathcal{V}_j^i$ we count, over the entire training set, how many times it contains less than four points and remove those $\mathcal{V}_j^i$ from the corresponding $\mathcal{T}_j$ for which this applies in at least 30% of the cases. The threshold comes from the definition of the features whose computation is undefined or poorly conditioned with four points. Voxels that are discarded in this way encode a non-informative location or scale for the characterization of the object. Examples include small voxels around the head, which is smaller in diameter than other body regions.

IV. Dataset

For our experiments, we use the SRL Human Attribute dataset [1] which contains 118 distinct persons (54 male, 64 female) annotated with gender and age in 137 different recordings. The data has been collected at 15 Hz in three different indoor locations under controlled lighting conditions using a Kinect v2 sensor. The subjects perform four different standing and walking patterns designed to cover all relative orientations and an RGB-D sensor range between $0.5m$ and $4.5m$. In sequence 1, the subject is standing at around $2.5m$ distance from the sensor and rotates clockwise in $45°$ steps (1 image per step). Sequence 2 consists of a video of the person performing a complex walking pattern. In sequence 3, the person walks on a circle that covers almost the entire view frustum. Finally, sequence 4 simulates a close-up interaction with a robot, where the subject steps back, forth and sideways in front of the sensor as if he/she is physically interacting with the robot's touch screen or manipulator. In total, these sequences contain around 1000 RGB-D frames per person.

For the experiments conducted in this paper, we annotated the persons with additional binary attributes *wears long trousers*, *has long hair*, *has long sleeves*, *wears jacket*. These attributes are per-person and do not vary across frames. Each subject has been independently annotated by three persons and in case of conflicting annotations, we chose the majority vote. This happened e. g. when a person was wearing $^3/_4$ trousers (which are not clearly *long trousers*, but also not shorts) or with medium-length hair. The absolute frequencies of these attributes across the dataset can be seen in the first three columns of the table in Fig. 3.

V. Experiments and Results

In our experiments, we compare the baseline method from our previous work on gender classification [1], which relies only on the features in Table I, against the new version extended with geometric extent features, color features and point cloud scaling. As opposed to [1], to limit training time we only use 100 instead of 500 weak classifiers. We also compare against a linear SVM classifier baseline trained on

	n_{pos}	n_{neg}	[1]	+Extents	+Scaling	+RGB /	HSV /	$\text{Y}'\text{C}_\text{B}\text{C}_\text{R}$	HOG
gender (m/w)	68	69	89.4%	89.0%	**91.7%**	90.2%	90.1%	91.3%	85.2%
long trousers	111	26	73.9%	72.6%	73.9%	80.8%	85.0%	**85.7%**	70.6%
long sleeves	60	77	63.9%	65.2%	63.6%	65.5%	71.6%	**73.2%**	69.8%
long hair	70	67	85.1%	84.0%	**87.7%**	86.8%	87.1%	86.2%	83.4%
jacket	20	117	62.8%	**63.8%**	61.4%	61.8%	57.4%	59.9%	56.5%
random label	68	69	50.7%	47.8%	50.5%	50.5%	51.6%	50.6%	49.7%

Fig. 3. Influence of point cloud scaling (Sec. III-D) and the additional geometric extents and color features in three different color spaces (all in Table II) on classification accuracy in sequence 1 (static poses). The second and third column show the number of positive and negative class instances per attribute. The fourth column shows the performance of our baseline method using only the geometric features from Table I. Our approach, using $n_{\text{weak}} = 100$, outperforms the HOG baseline in both accuracy and runtime performance, where our method is around $15\times$ faster on a single CPU core.

HOG features in RGB-D. While we mainly focus on human attributes that are reasonably well presented in the dataset, we believe that in principle our algorithm is applicable to a broader range of appearance-based human attributes (e. g. *has hat*, *has backpack*).

Our tessellation-based classifier is implemented in C++ and integrated with ROS for visualization. We use the AdaBoost implementation from the OpenCV library, as well as the Point Cloud Library (PCL) to load and pre-process pre-extracted person clouds. Feature computations are parallelized using OpenMP. Our code will be released to the public upon publication of this paper.

For the HOG baseline, we compute HOG feature descriptors using the OpenCV library on both the RGB and depth image of the person with a window size of 64×128 pixels, which in previous experiments gave better results than 32×64 px [1]. The resulting feature descriptors in RGB and depth are concatenated and then fed into a linear SVM.

To maintain a reasonable size of training and test sets during cross-validation, we perform 10 rounds of repeated random sub-sampling validation. In each round, we randomly divide the dataset on a per-person basis into two approximately equally sized training and test sets. We take measures to ensure that one person instance never appears in the training and test set simultaneously to prevent the classifier from learning individual persons' appearances. We then average the classification accuracy on the test sets over all 10 folds. To keep training times within reasonable limits, we subsample the frames of the larger sequences 2–4 of the dataset by a factor of 5. For class balancing, we conduct undersampling on a frame-by-frame basis by discarding excess samples of the majority class, such that each train and test set eventually contains 50% positive and 50% negative sample frames.

A. Classification accuracy

Static poses only: Results in Fig. 3 compare the different extensions discussed in Sec. III against the baseline methods on seq. 1 of the dataset – containing only standing persons – over different human attributes. In the last row, each person instance has been assigned a random label, thus the expectation here is that the classifier should not behave significantly better than chance (50%). We can see that, in all cases, our extensions improve classification accuracy. $\text{Y}'\text{C}_\text{B}\text{C}_\text{R}$ features are especially useful for detecting long sleeves and trousers, which is expected since the absence of those is indicated by an increase in visible skin color. As expected, color features are not so helpful for detecting gender and long hair. It can also be seen that in a number of cases, the inclusion of further attributes leads to reduced test performance, e. g. when adding RGB or HSV color features to the gender or jacket classifier. As we often achieve an accuracy of 90 to 100% on the corresponding training set, we are confident that this is a sign of overfitting to the training data. A significantly larger training set (which, in RGB-D, is very expensive to acquire) should help to alleviate these effects.

As the attributes *long hair* and *gender* are highly correlated in the groundtruth (ρ=-0.90), we conduct an additional experiment to analyze if the *long hair* classifier still performs well when only the upper body including the head, but excluding the region below the waist – whose shape can be an important indicator for gender – is visible. To do so, we cut all point clouds below a fixed height of 1.2m. In this case, accuracy only drops from 87.7% to 86.1%. If we instead only consider the uppermost 0.35m of each point cloud, which contain the head, we obtain an average accuracy of 83.5%, which is still a good result, but at the same time visualizes how individual attribute classifiers can benefit from additional contextual shape information.

Full dataset: In Fig. 4, we show results when training and testing a classifier on the full dataset, including walking and close-up interaction sequences. Our approach outperforms the baselines in all cases except for the *has jacket* attribute, which is the most under-represented attribute in our dataset and probably suffers from overfitting on the color features. We also want to note that this is a difficult binary attribute to detect, as sometimes the subject is wearing a cardigan, which in its appearance ofteny resembles a jacket but is not annotated as such. For *long trousers* and *long sleeves*, we improve the accuracy by around 12% compared to the original method from [1]. As expected, the overall performance goes down by 3–5% under less controlled conditions when persons are walking instead of just standing (seq. 2–3), and in very close proximity to the sensor (seq. 4).

Gender	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG	78.0%	76.9%	77.0%	77.4%
[1]	89.8%	83.7%	82.6%	85.1%
Ours	**90.4%**	**87.0%**	**86.3%**	**87.7%**
Long trousers	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG	65.0%	60.0%	59.4%	60.7%
[1]	69.4%	66.0%	64.1%	67.0%
Ours	**83.6%**	**78.0%**	**76.2%**	**79.9%**
Long sleeves	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG	63.2%	60.8%	60.7%	61.9%
[1]	62.3%	61.8%	61.0%	61.8%
Ours	**76.9%**	**73.8%**	**72.8%**	**74.3%**
Long hair	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG	74.3%	72.6%	72.7%	73.3%
[1]	83.7%	77.9%	77.2%	79.3%
Ours	**87.2%**	**83.3%**	**82.9%**	**83.9%**
Jacket	(1)	(1)–(3)	(1)–(4)	d > 0.8m
HOG	56.7%	56.5%	56.8%	57.0%
[1]	**62.8%**	**62.3%**	**61.5%**	**62.1%**
Ours	60.8%	59.3%	59.0%	59.1%

Fig. 4. Classification accuracies for the full dataset, including static poses (seq. 1), walking sequences (seq. 2+3) and close-up interaction (seq. 4). 10 runs of repeated random sub-sampling validation with $n_{\text{weak}} = 100$, using extent and YCbCr color features and point cloud scaling. For HOG, we use a $64x128$ window size. The last column excludes all frames where the person is closer than 80 cm to the sensor, where only a very limited part of the body is visible and clipping artefacts sometimes appear in the data.

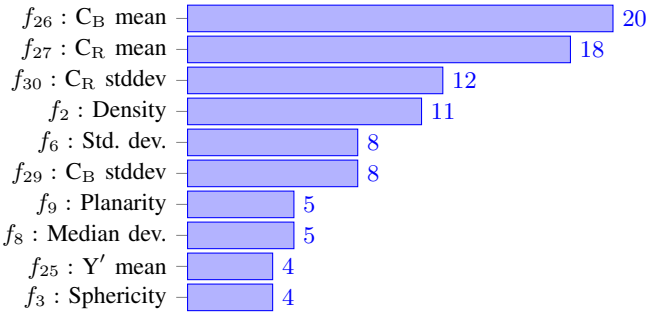

Fig. 5. Best 10 geometric and $Y'C_BC_R$ color features selected by our approach for the *long trousers* attribute on the full dataset. Numbers are absolute frequencies and are relative to the total number of 100 weak classifiers used during training.

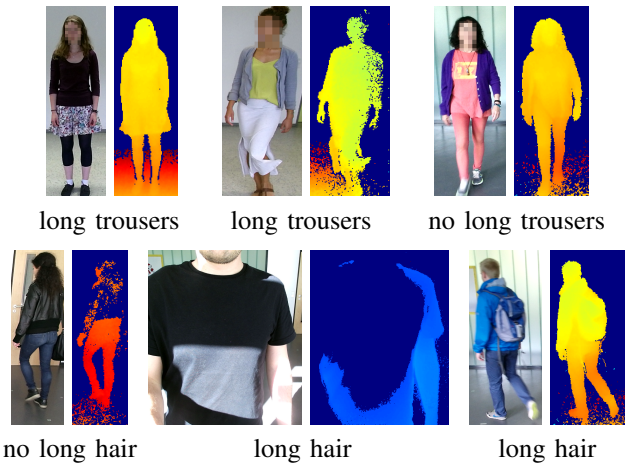

Fig. 6. Examples of typical failure cases for the *long trousers* attribute (top row) and the *long hair* attribute (bottom row). The caption below each RGB and depth image shows the predicted, incorrect class label.

B. Feature selection

At training time, our method computes for *all* voxels of *all* tessellations *all* features referred to by the respective column in Fig. 3. For instance, for the second-last column, the features (1)–(9), (10)–(12), (25)–(30). This leads to a very high-dimensional feature vector. During testing, however, only the *most informative* features in the most informative voxels of all tessellations are calculated; that selection of features is implicitly given by the best n_{weak} weak classifiers chosen by Adaboost.

To find out how often the proposed new features are selected by our boosting approach overall, we count the absolute usage frequency of each feature type across all 100 weak classifiers. Fig. 5 shows the 10 most frequently used features for the *long trousers* attribute. It can be seen that the means of the chroma color channels are being used fairly often, but geometric features such as density, standard deviation w.r.t. centroid and planarity still play a large role. Also taking other human attributes into account, we note that standard deviations in the color channels are being used less often than the corresponding means, and that the geometric extent features (width, depth, height) are used less than 5 times on average across all human attributes.

C. Failure cases

In Fig. 6, we show example RGB and depth images depicting cases where our approach typically fails. The first row shows the *long trousers* attribute; here, we notice that our classifier sometimes fails to detect very thin trousers, such as panty hoses, that do not protrude in 3D (first image), or when the shape of the clothing is not well-defined and changes dramatically during motion (middle image). Also, nearly skin-colored trousers (last image) are not detected as such, which we believe is due to this trousers color not otherwise appearing in the training set. For the *long hair* attribute (second row), the classifier may fail when depth data is totally missing due to clothing surface properties that irritate the RGB-D sensor (first image), when the person is too close to the near clipping plane such that a significant part of the point cloud is missing (middle image), or when accessories like hoods or backpacks generate unusual shapes in the point cloud which are significantly under-represented in our training set (last image).

D. Computational efficiency

Voxel filtering: Pruning mostly empty voxels from the set of tessellations before training as described in Sec. III-D reduces memory consumption significantly from 12 GB to around 2.4 GB (-80%) on the full dataset using just the geometric features. Due to the otherwise extremely high-dimensional feature vectors, this pre-processing step becomes even more important when additionally including color features. While testing on seq. 1, no negative impact on classification accuracy was observed when including this additional filtering step. Although the pre-filtering incurs some processing time overhead, this is remedied by the fact that the feature vectors fed into the AdaBoost learning algorithm become significantly shorter.

Training time: Using 4 parallel threads for feature computations, training on a single randomized 50% subset of the full dataset (while using every 5^{th} frame of seq. 2–4) takes between 0.5 and 2 hours for a single human attribute, depending on the number of features being used.

Runtime performance: Efficient runtime performance is important for resource-constrained mobile service robots. Since a single scene might contain multiple persons, each with multiple attributes to detect, the processing time per input person cloud should be low. With $n_{\text{weak}} = 100$ and using 4 threads, our classifier runs in real-time at around 300 Hz (for a single attribute) on pre-extracted and aligned RGB-D person point clouds without requiring GPU acceleration. Using just a single thread, we still achieve about 125 Hz. If we instead use $n_{\text{weak}} = 500$ as in [1], accuracies improve by around 1-3% while still allowing processing at 120 Hz (4 cores) or 40 Hz (1 core). The new features added in this paper do not have a significant impact on performance, as we selectively only compute the best features (found by AdaBoost) in each selected voxel, and all features are very simple to compute. Also, adding additional features at training time does not increase the feature vector size during testing as long as the number of weak classifiers remains constant.

VI. Conclusion

In this paper, we extended our existing approach on full-body human gender recognition using a tessellation boosting approach [1], which so far only used 3D information, with color features and two new pre-processing steps. These extensions lead to an improved classification accuracy, shorter training times and lower memory consumption. Furthermore, we conduct experiments on four additional, challenging human attributes and obtain first promising results especially for the attributes *long trousers* and *long hair*. Our method is very efficient, achieving classification rates per attribute of up to 300 Hz, which is important for resource-constrained mobile service robots that are supposed to learn more about their human environment.

In future work, we want to integrate our classifier with a real-time RGB-D people detection and tracking framework on a mobile robot and smooth the predicted class labels over time to overcome occasional misclassifications due to, for example, extreme body poses that do not appear in our training data. We also plan to extract additional training data from a dataset recorded in a crowded pedestrian environment in order to improve performance on the existing human attributes and to learn new ones such as age group or the presence of backpack or luggage. Finally, we intend to learn and combine multiple height-specific classifiers in order to boost classification accuracy e. g. for children.

References

[1] T. Linder, S. Wehner, and K. O. Arras, "Real-time full-body human gender recognition in (RGB)-D data," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[2] W. Liu, T. Xia, J. Wan, Y. Zhang, and J. Li, "Rgb-d based multi-attribute people search in intelligent visual surveillance," in *Advances in Multimedia Modeling*, ser. Lecture Notes in Computer Science, K. Schoeffmann, B. Merialdo, A. Hauptmann, C.-W. Ngo, Y. Andreopoulos, and C. Breiteneder, Eds. Springer Berlin Heidelberg, 2012, vol. 7131, pp. 750–760.

[3] M. Munaro, A. Fossati, A. Basso, E. Menegatti, and L. Van Gool, "One-shot person re-identification with a consumer depth camera," in *Person Re-Identification*, ser. Advances in Computer Vision and Pattern Recognition. Springer, 2014.

[4] I. B. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino, "Re-identification with RGB-D sensors," in *ECCV 2012 Workshops and Demonstrations*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7583.

[5] H.-J. Wang, Y.-L. Lin, C.-Y. Huang, Y.-L. Hou, and W. Hsu, "Full body human attribute detection in indoor surveillance environment using color-depth information," in *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, Aug 2013, pp. 383–388.

[6] H.-C. Lian and B.-L. Lu, "Multi-view gender classification using local binary patterns and support vector machines," in *Advances in Neural Networks (ISNN 2006)*, ser. Lecture Notes in Computer Science, 2006, vol. 3972.

[7] B. Li, X.-C. Lian, and B.-L. Lu, "Gender classification by combining clothing, hair and facial component classifiers," *Neurocomputing*, vol. 76, no. 1, 2012.

[8] C. Shan, "Learning local binary patterns for gender classification on real-world face images," *Pattern Recognition Letters*, vol. 33, no. 4, 2012.

[9] M. Castrillón-Santana, J. Lorenzo-Navarro, and E. Ramón-Balmaseda, "Improving gender classification accuracy in the wild," in *Proceedings of the 18th Iberoamerican Congress on Pattern Recognition (CIARP 2013)*, ser. LNCS, vol. 8259, 2013.

[10] J. Tang, X. Liu, H. Cheng, and K. Robinette, "Gender recognition using 3-d human body shapes," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 41, no. 6, 2011.

[11] L. D. Bourdev, S. Maji, and J. Malik, "Describing people: A poselet-based approach to attribute classification," in *IEEE Int. Conf. on Comp. Vis. (ICCV)*, 2011.

[12] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. D. Bourdev, "PANDA: pose aligned networks for deep attribute modeling," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 1637–1644.

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