Real-time Person Orientation Estimation using Colored Pointclouds

Tim Wengefeld, Benjamin Lewandowski, Daniel Seichter, Lennard Pfennig and Horst-Michael Gross

***Abstract*— Robustly estimating the orientations of people is a crucial precondition for a wide range of applications. Especially for autonomous systems operating in populated environments, the orientation of a person can give valuable information to increase their acceptance. Given people's orientations, mobile systems can apply navigation strategies which take people's proxemics into account or approach them in a human like manner to perform human robot interaction (HRI) tasks. In this paper, we present an approach for person orientation estimation based on performant features extracted from colored point clouds, formerly used for a two class person attribute classification. The classification approach has been extended to the continuous domain while treating the problem of orientation estimation in real time. We compare the performance of orientation estimation treated as a multi-class as well as a regression problem. The proposed approach achieves a mean angular error (MAE) of $15.4°$ at 14.3ms execution time and can be further tuned to $12.2°$ MAE with 79.8ms execution time. This can compete with accuracies from state-of-the-art and even deep learning based skeleton estimation approaches while retaining the real-time capability on a standard CPU.**

I. INTRODUCTION

The current orientation of persons (see Fig. 1) in the surroundings of a robot is a useful attribute for various HRI tasks. In the field of socially aware robot navigation, mainly two core functionalities require orientation information. First, for approaching a person correctly, the orientation of the user provides useful information for positioning in order to allow an unconstrained interaction. Second, in proxemic theory the orientation is used to model a natural personal space around a person which the robot should not enter during the navigation process. However, in such high-level applications, the orientation is typically considered to be given, either through motion capture data [1] or other external sensor systems [2], which cannot be used in real world scenarios. In spacious application areas, the sensor setup is limited to the mobile robot platform. Some approaches use closed source 3D skeleton estimators like the OpenNI- or Kinect2-SDK in order to derive orientation information. These in turn have been used for approaching [3, 4] or personal space [5] applications. This might work in an experimental setup, but several limitations restrict the fields of application. For example, the constrained detection space of usually $1.5\mathrm{m} - 4.5\mathrm{m}$ limits the navigation planning horizon.

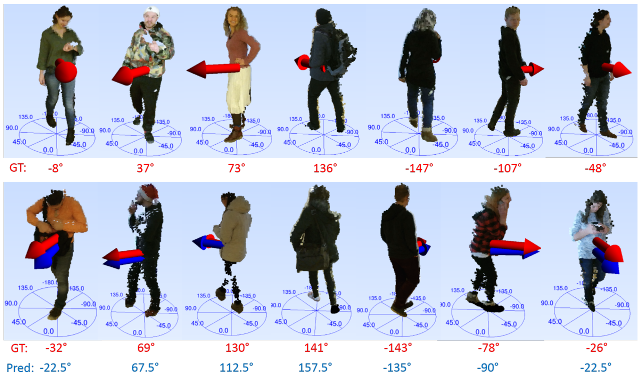

GT: -8° 37° 73° 136° -147° -107° -48°
GT: -32° 69° 130° 141° -143° -78° -26°
Pred: -22.5° 67.5° 112.5° 157.5° -135° -90° -22.5°

Fig. 1: Various samples from our train-set (top) and test-set (bottom). The ground truths (red) and predictions (blue) are indicated with arrows.

Other specifics, like a prior on the orientation of users with respect to the sensor's coordinates as seen in the Kinect2-SDK, can lead to failures in case of approaching a person, not facing the sensor.

Person orientation estimation itself was undergoing extensive studies. However, for the best of our knowledge, all robotic applications which use these information fall back to more or less restricted methods not generally applicable in a real-world scenario. Therefore, we consider the robust and fast estimation of a person's orientation still as a challenging field for research.

The main contributions of this paper are as follows:

1) We provide a fast and accurate approach for person orientation estimation[1], which exploits rich information of colored point clouds. This approach can even compete with the accuracy of recent deep learning methods but also allows computation in real-time on a standard CPU without the need of specialized graphics cards, as we will show in the experimental section.
2) We perform an extensive evaluation of the precision of our approach on a self recorded dataset[6] with over 100,000 samples of colored point clouds and compare the achieved accuracies of multi-class classification and regression.
3) Since our approach is flexible with respect to estimation accuracy and computation time, we compare different levels of complexity to serve (i) applications with a high need of accuracy but a lower time constraint, like approaching static persons, and (ii)

All Authors are with Neuroinformatics and Cognitive Robotics Lab, Technische Universitaet Ilmenau, 98694 Ilmenau, Germany. tim.wengefeld@tu-ilmenau.de

This work has received funding from the German Federal Ministry of Education and Research (BMBF) to the project 3D-PersA2 (grant agreement no. 03ZZ0460), and to the project ROTATOR (grant agreement no. 03ZZ0437D) both in the program Zwanzig20 Partnership for Innovation as part of the research alliance 3DSensation.

[1] The code is publicly available at https://github.com/TimWengefeld/OrientationEstimation

978-1-7281-3605-9/19/$31.00 ©2019 IEEE

Europ. Conf. on Mobile Robotics (ECMR), Prague, Czech Republic, 7 pages, IEEE 201916/12

applications where a fast prediction is more important than a high accuracy, like personal space estimation or person tracking.

II. Related Works

Person orientation estimation, as well as the similar problem of estimating the head orientation of a person, is usually treated by one of two estimation strategies. The first strategy handles the estimation problem as a multi-class classification by discretizing the continuous prediction space into several orientation classes. In contrast to that, more recent works perform a regression by predicting the continuous angle directly. In [7], a combined detection and orientation estimation approach is presented using Histogram of Oriented Gradient (HOG) features and a Decision Tree of Support Vector Machines to detect eight upper body orientation classes and one background class. In [8], different combinations of well known RGB feature descriptors like HOG, LBP and ACF were used for an eight class orientation estimation. [9] extends the HOG feature space with the magnitude of gradients from depth images, which increases the feature weights at the boundary of the human body silhouette. In this way, they achieved better performance on data with complex background. All of these traditional machine learning approaches have in common, that they divide the prediction space into relatively coarse classes. This comes with the drawback, that in addition to misclassifications (reported accuracies range between 40% and 80%), a systematical error is introduced, which is about 11.25° considering a balanced test set and eight orientation classes. Similar to other domains, recent advances in deep learning have improved the accuracy of orientation estimation approaches significantly. One of the first deep learning approaches for orientation estimation [10] uses a deep convolutional neural network (CNN) with cropped and resized person appearances from RGB-images as input and a softmax layer with eight neurons as output, which gives the confidence for each of the eight trained classes. They report an mean angular error (MAE) of 10.6°. However, their evaluation method also neglects the mentioned systematical error, so the MAE must be higher in reality. In [11], a regression approach is presented which estimates the head orientation in RGB-images with a deep CNN using two output neurons trained on the sine and cosine part of the prediction angle. They reported a MAE of 20.8° on a real-world dataset recorded in a town center. A similar approach presented in [12] predicts the full body orientation using cropped and resized greyscale images of people as input and two neurons as output. The CNN was trained with a large synthetic training set and the evaluation was done on a spinning wheel which can measure the ground truth with a non changing simple background. Astonishingly good results of 6.9° MAE are reported on this relatively simple test set. Another way to estimate a person's orientation is to retrieve this information from estimated skeletons. In the past few years, this field of research has produced outstanding results. [13] as probably the most famous representative, commonly known as OpenPose, predicts 2D skeletons in RGB-images.

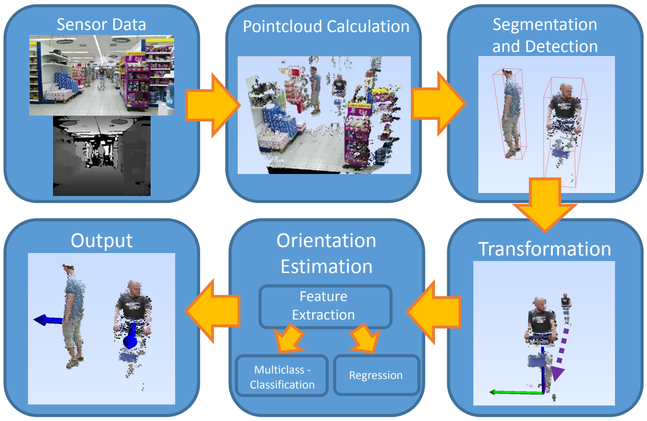


Fig. 2: Processing pipeline of the proposed approach in a supermarket scenery.

However, from these 2D information the person's 3D orientation cannot be retrieved directly. Other recent approaches are able to estimate 3D skeletons from RGB[14] and RGB-D[15] images. We compare our approach to them in the experimental section. However, when it comes to the application, not every problem can be treated with deep learning due to the need of graphics cards which opposes to the restricted resources on mobile platforms. Even though there are power-saving graphics cards available, like the NVIDIA Jetson series, complex robotic systems [16, 17] which also need person re-identification or scene understanding benefit from computationally less expensive alternatives for specific tasks.

III. Robotic Application

The proposed approach for orientation estimation is based on the attribute estimation presented in [18], which uses clusters of colored 3D points as input. There, person point clusters are generated within a static sensor setup using a background model. In order to use this approach on a mobile platform, where background models are not feasible, we integrated it in a processing pipeline using the robotic middle ware framework MIRA [19] (see Fig. 2). Color and depth images from a Kinect2 sensor are transformed into a colored point cloud. Afterwards, the segmentation method from [20] is applied to generate candidate point clusters which possibly represent persons. For cluster validation, we use the person detector presented in [21], which currently delivers the best detection results for this data representation. The advantage of this and other point-cloud based person detectors [20, 22] is that they share the required segmentation step with our approach and, therefore, no computational overhead is generated when used in combination. But in general, arbitrary person detectors combined with a projection into the point cloud coordinate system could be used to validate that a point cluster originates from a person. Another advantage of using clusters of point clouds as data representation is the independence from different backgrounds. For detection accuracies in a supermarket scenario, we refer to [21]. After the detection step, the person point clusters are transformed

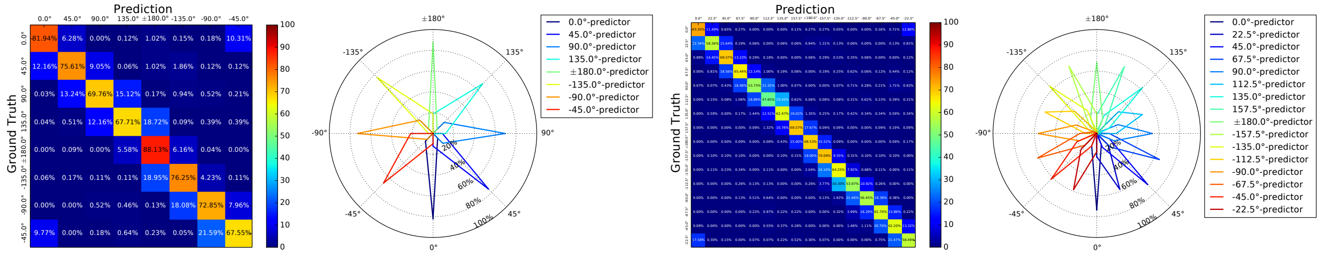


Fig. 3: Results for proposed orientation classifiers for the 8 class CV-SAC-WL100-D1-CC8 (MAE: 17.84°) classifier (left) and the 16 class CV-SAC-WL500-D2-CC16 (MAE: 12.21°) classifier (right). The confusion matrices are shown on the left. They indicate how likely each binned ground truth label is predicted to each orientation class. These results are further transformed into a radial coordinate system shown on the right to get a more intuitive representation. In this plot, the radial dimension encodes the prediction probability while the circular dimension encodes the ground truth label.

in a local coordinate system aligned at the cluster's center of gravity. This normalized point cloud representation of each detected person is passed to the orientation estimation module, described in the following section.

IV. ORIENTATION ESTIMATION

As mentioned before, the proposed method is a modification of the human attribute classification approach from [18] which classifies binary attributes like *gender* or *long/short trousers*. In this former work, a typical Adaboost approach has been used where a small amount of simple, yet fast to calculate features are extracted from a large amount of different regions of a colored person point cloud to train the classifier. More precisely, they used 19 statistic, geometric and color features, like the *number of points*, *linearity* and *mean color*, from 14,023 overlapping subregions of the point cloud. The main idea of such boosting approaches is, that the designer of the algorithm does not have to put much effort into data preprocessing, feature design, or feature selection, since the Adaboost training aims to find the most distinctive features by itself. An Adaboost classifier can be described in the notation of eq. 1. For a feature vector x, an ensemble of T consecutively trained *weak classifiers* $f_t(x)$ (typically decision trees) vote with a confidence for a class label. This predicted confidence is then multiplied with a scalar factor α_t which encodes the importance of the specific *weak classifiers* decision to the combined decision for the *strong classifier* $F(x)$. For more details, we refer to the original Adaboost paper [23].

$$F(x) = \sum_{t=1}^{T} \alpha_t f_t(x) \tag{1}$$

The main advantage of this procedure is, that the prediction step of the classifier is quite fast since in the application phase only important features need to be calculated. This makes these approaches real-time capable with a low need of computational resources. However, the original Adaboost algorithm from [23] is just able to solve binary classification problems.

A. *Orientation estimation by Multi-class classification*

A typical approach to overcome this issue is to train several binary classifiers and make a one-vs.-all decision based on the maximum classification confidence of each classifier:

$$max\,\{F_{0^\circ}(x), F_{45^\circ}(x), ..., F_{-45^\circ}(x)\}$$

However, dividing the former continuous prediction space into discrete bins leads to the issue that samples which are close to each other in the feature space may fall into different classes. Furthermore, if we take noise in our ground truth labels into consideration, which is very likely when training on data which is not artificially generated, one classifier may be trained with similar samples which have both positive and negative labels. This is an issue often ignored in most state-of-the-art publications. To the best of our knowledge, the only approach which treats this issue was presented in [24]. There, the SVM training algorithm was extended with a cost relaxation parameter which weights the errors of similar classes lower than fatal misclassifications. Using this extension, they achieve an accuracy improvement for the eight class classification of up to 3.23%. However, the fundamental idea of the Adaboost algorithm is to find importance weights for specific samples by itself during training. Hence, this method cannot be adapted without changing this principle of Adaboost. Therefore, we simply tackled this issue by just ignoring samples adjacent to the positive class in the training step of a classifier for a specific direction. In the experimental section, we will show that we can achieve a notable precision boost with this simple yet efficient training strategy. Furthermore, this training strategy allows us to choose more fine granulated subdivisions of our prediction space than the de facto standard of eight classes to reduce the MAE, while a normal one-vs-all training just increases the MAE for a finer granulated prediction space division.

B. *Orientation estimation by Regression*

Another strategy is to treat the orientation estimation as a regression problem. The Adaboost algorithm was generalized to regression problems in [25] calling it gradient boosting.

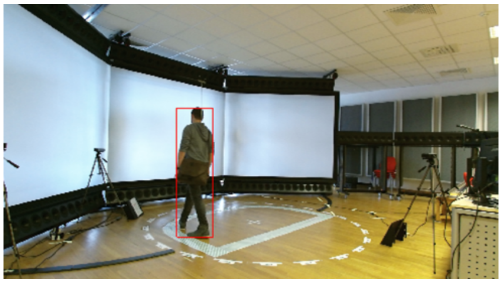
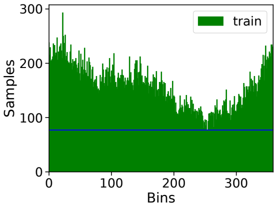


Fig. 4: Environment where we recorded our dataset.



dataset	# samples
train-full	57,717
train-balanced	27,720
test-full	50,788
test-balanced	21,600

Fig. 5: Left: histogram of the orientation angles in our recorded training set. The blue line indicates the number of samples drawn for balancing. Right: statistics of the datasets.

The concept is pretty similar to the original Adaboost by consecutively training weak classifiers, also typically decision trees, which solve the problem by finding the optimal features. However, instead of independently returning a class confidence, the first weak classifier predicts the mean of the real-valued labels, while consecutive ones iteratively minimize the residual errors from previous steps during the training. In the application phase, these weak classifiers can be evaluated independently giving this machine learning approach the same fast prediction capabilities. However, treating the orientation estimation as a regression problem comes along with the problem of periodicity of angles. This leads to the effect, that two samples close to each other in reality would have a high distance for our model if they are on different sides at the transition point of our prediction space, i.e. $-179°$ and $+179°$. In general, such issues can be handled by transforming the prediction label to a higher dimensional space. However, gradient boosting classifiers are natively not capable to provide multi-dimensional outputs. In [11] the periodicity of the orientation angle is treated by a CNN with a two dimensional output for the sine and cosine part of the prediction angle. In our approach, we have adapted this method and trained one separate model for the sine and cosine component of the angle.

V. EXPERIMENTS

Over the years, several evaluation metrics have been developed. Most classification approaches use the accuracy evaluation metric for correctly classified samples. However, since we are just interested in a precise estimation of the real valued orientation in the application, we will use the mean angular error (MAE) evaluation metric independently from class division for classification or regression. The MAE represents the mean absolute difference between ground truth and predicted angles while taking the circularity of the prediction space into account. Computation times are averaged results per sample over our balanced test-set on an Intel Core i7-4790K using 4 cores.

A. Datasets

In the literature, a considerable amount of data sets for training and benchmarking orientation estimation of persons are publicly available with RGB [26] and RGB-D [27] data. However, to the best of our knowledge, none of them fulfill our requirements of synchronized depth and RGB data streams to generate point clouds in combination with highly precise ground truth labels suitable for both classification and regression. Therefore, we decided to record our own data set using a highly precise external ARTTRACK tracking system [28], which tracks markers using four infrared (IR) cameras with a positional precision of $0.4\text{mm} \pm 0.06\text{mm}$ [29]. In order to reduce the influence of the markers in the actual point cloud data, the IR markers were placed on a thin pole about 0.5m above the heads (see Fig. 4). The pole itself was fixated under the cloth at the back of the subject to keep the orientation label unaffected from the head/view direction. By means of that, the label reflects the orientation of the person's thorax. We recorded data from five Microsoft Kinect2 sensors placed in a half circle around the recording area such that the sensors' active boosters did not interfere with each other. Different behaviors of daily life, like walking around, using cellphones, or conversations were captured. To generate samples to train our approach and to evaluate the performance we used a background model. Afterwards, we applied the person detection pipeline described in Sec. III to filter noise and to use the same preprocessing as in the application phase. During the sessions, appearances of 37 persons were recorded in a range of 1.5m to 5m which we divided into groups of 21 persons for training and 16 for the test set. To give a valid evaluation of the system's generalization capabilities, no person is included in both sets. We also tried to keep the variance of persons' attributes in the test set high with respect to gender, height, and clothing (see Fig. 1). During the complete session 57,717 samples for the training set and 50,788 samples for test set were recorded. Unfortunately, since we placed the sensors in a half circle, some of the recorded ground truth angles are overrepresented. Therefore, we balanced the data sets by sampling from 360 angle bins randomly without using a sample twice. The resulting data sets contain 27,720 training samples and 21,600 test samples (see Fig. 5). In the following, we discuss the results on the balanced test set only. Results for the full test set are given in the Tab. I and II, but are not discussed.

B. Classification results

One of the first issues we faced during the performance evaluation of our approach was the computational effort of the Adaboost algorithm during training. Given a large dataset with several parameter configurations and an increasing

exp. series	Identifier	test MAE bal.	test MAE full	avg. time feat.	avg. time class.	#WL per class.	#class	tree dep.	#train samples
1	CVAB-UAC-WL100-D1-CC8	20.92°	20.57°	7.89ms	0.03ms	100	8	1	8,000
	CVAB-SAC-WL100-D1-CC8	17.84°	17.51°	6.36ms	0.03ms	100	8	1	8,000
	CVAB-UAC-WL100-D1-CC16	22.45°	21.93°	13.78ms	0.09ms	100	16	1	8,000
	CVAB-SAC-WL100-D1-CC16	15.40°	14.78°	14.30ms	0.09ms	100	16	1	8,000
	CVAB-UAC-WL100-D1-CC32	24.66°	23.49°	29.39ms	0.22ms	100	32	1	8,000
	CVAB-SAC-WL100-D1-CC32	21.18°	19.68°	29.12ms	0.23ms	100	32	1	8,000
2	CVAB-SAC-WL100-D2-CC16	14.31°	13.76°	28.73ms	0.24ms	100	16	2	8,000
	CVAB-SAC-WL100-D3-CC16	26.38°	25.86°	24.06ms	0.15ms	100	16	3	8,000
	CVAB-SAC-WL500-D2-CC16	12.21°	11.80°	78.96ms	0.96ms	500	16	2	8,000

TABLE I: Results of our classification approach with different training parameters in context of mean angular error (MAE) and computation time for feature calculation and classification. The identifier encodes the parameters the multi-class classifier was trained with in the form (machine learning back end [Open**CV** **A**da**B**oost] - training strategy (**U**se/**S**kip **A**djacent **C**lasses) - # weak learners per classifier - tree depth - # prediction classes).

exp. series	Identifier	test MAE balanced	test MAE full	time feat	time class	#WL per classifier	tree depth	#train samples
1	CVGBT-WL800-D1	17.68°	17.27°	5.36ms	0.15ms	800	1	8,000
	CVGBT-WL800-D2	15.17°	14.76°	7.80ms	0.13ms	800	2	8,000
2	XGB-WL800-D2	12.55°	12.24°	*	0.60ms	800	2	27,720
	XGB-WL3200-D3	11.52°	11.21°	*	0.70ms	3200	3	27,720

TABLE II: Results of our regression approach with different training parameters in context of mean angular error (MAE) and computation time for feature calculation and classification. The identifier encodes the training parameters of the classifier in the form (machine learning back end [Open**CV** **G**radient **B**oosted **T**rees / **XGB**oost] - # weak learners per classifier - # tree depth. (*) Results for XGBoost are currently just retrieved from the python interface with pre-calculated features.

amount of prediction classes, the training can easily exceed several weeks using the OpenCV [30] machine learning back end. Therefore, the training was not possible on the full balanced training set and thus, it had to be reduced to 8,000 samples for the parameter evaluation, which also needs our maximum amount of 32GB RAM. For accuracy evaluation in order to find the best training parameters, we conducted two series of experiments. The first one is intended to prove the advantage of our training strategy and to find the best number of subdivision for the continuous prediction space, i.e. the number of classes. In the second series, the accuracy of the best performing method should be increased by variating the training parameters, i.e. number and tree depth of the weak learners. For the complete parameter configuration, we refer to the implementation[2].The results are shown in Tab. I. As expected, the SAC (skip adjacent classes) training strategy performs better than the standard approach using all samples. Therefore, we used this strategy in the following experiments. The first classifier trained with eight classes performed surprisingly well by predicting the orientation with an MAE of 17.84°[3] with an average execution time of 6.39ms per sample. The specific results for each orientation can be found in Fig. 3. There, it becomes obvious (especially in the polar plots), that frontal or backward appearances can be estimated more precisely than sideviews. This can be explained with a larger surface of the persons in such appearances and more descriptive features resulting from them. However, none of the ground truth classes is estimated extremely worse than others and mispredictions are mostly adjacent to the real ground truth rather than pointing into complete different directions. By increasing the number of prediction classes to 16 the MAE drops to 15.40° with an execution time of 14.39ms. However, setting the number of classes to 32 performs worse than 16 but better than eight classes. Therefore, we assume 16 classes to be the optimal subdivision of our prediction space. In our second experimental series, we performed a parameter grid search over the number of weak learners (ranging from 100 to 500) and their maximum tree depth (ranging from 1 to 3). Exemplary results can also be found in Tab. I (for all results we refer to our github repository). The best combination achieves 12.21° MAE for our classification approach using 16 classes, 500 weak learners, and a weak learner's maximum tree depth of two. However, increasing the number of classes or model complexity comes with the drawback of a higher computation time of up to 79.92ms per sample. Thus, the optimal model has to be chosen for each application separately. It depends on whether accuracy or low latency is more important. Since we are just interested in real-time predictions, this is the maximum tolerable execution time for our application, where typically more than one person appears in a scene. In the following, we will refer to the CVAB-SAC-WL100-D1-CC16 classifier with 15.40° MAE and 14.39ms execution time as *ours-classification-fast* and the CVAB-SAC-WL500-D2-CC16 classifier with 12.21° MAE and 79.92ms execution time as *ours-classification-precise*.

[2] `https://github.com/TimWengefeld/OrientationEstimation`

[3] The best achievable accuracy for an eight class classifier due to discretization is an MAE of 11.25° given a well balanced test set.

Approach	test MAE bal.	avg. comp. time (orientation est.)	avg. comp. time (detection)
[14]	22.60°	1127.48ms*	
[15]	14.66°	606.33ms*	
ours-classification-fast	15.40°	13.7ms$^\bullet$	72ms$^\bullet$
ours-classification-precise	12.21°	105.9ms$^\bullet$	
ours-regression-cv	15.17°	20.8ms $^\bullet$	
ours-regression-xgb	11.52°	-	

TABLE III: Comparison of our approaches to two recent deep learning 3D skeleton estimation approaches. For runtime comparison, we applied the deep learning approaches on the Jetson Xavier(*), NVIDIAs most recent graphics card designed for mobile autonomous systems. For our approach, we measured the average runtime on our robot's i7-7700T($^\bullet$) CPU, using four threads.

C. Regression results

By using the same OpenCV machine learning back end for gradient boosting for regression, the same limitations considering the training set size hold true. In the first experimental series (see Tab. II), we also performed a parameter grid search over the number of weak learners (ranging from 100 to 800) and their maximum tree depth (ranging from 1 to 3). The best OpenCV regressors (CVGBT-WL800-D2 in the following referred as ours-regression-cv) achieved a MAE of $15.17°$ with a computation time of 7.80ms, while a classification approach with similar time consumption performs $2.67°$ worse. However, we have not achieved the same accuracy of the best multi-class classifier yet due to the large amount of training times. But since the regression approach is faster in general, the training of more complex models seems to be a valid option for future work experiments. Our second experimental series shows the reachable accuracy of the regression approach when using the complete training date available. Therefore, we changed to the more recent machine learning back-end for gradient boosting XGBoost [31], which provides better parallelization support and a more efficient memory management. With the same training parameters, like the best OpenCV regressor, the MAE decreased from $15.17°$ to $12.64°$. We are currently not able to give a run time comparison for XGBoost since we need to re implement the optimized feature calculation for this back end. But since feature calculation is the computational bottleneck, the run time should be similar to the OpenCV results with equal model complexities. With XGBoost as back end and a parameter grid search (see Fig. 6) over the number of weak learners (ranging from 200 to 3200) and their maximum tree depth (ranging from 1 to 3), our best regression approach (see Fig. 7) achieved a MAE of $11.52°$ (XGB-WL3200-D3 in the following referred as ours-regression-xgb) which is $0.69°$ better than our precise classification approach. Moreover, we analyzed the model complexity and the dependency of the model performance to the amount of samples used during training (see Fig. 6). There, it can bee seen that more complex models perform better in general. The enlargement of training set size from 14,400 to 27,700 samples gives a performance boost of about $1°$ MAE. However, increasing the model complexity above a tree depth of two and 1,600 weak learners just give a minor performance boost. Therefore, we conclude that our approach is near the maximum of its achievable accuracy. Enhancing the training set with further samples could give a slight performance boost, but for more precise estimations more complex features are required.

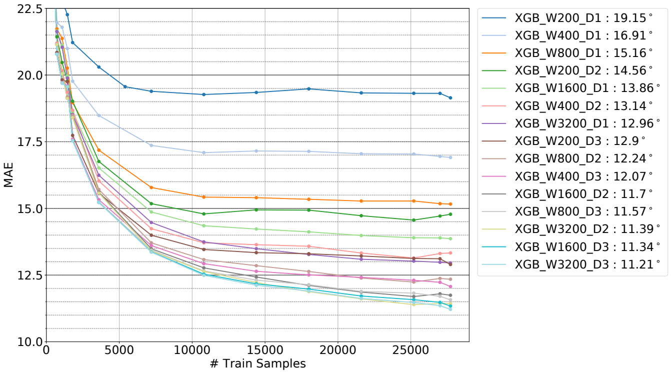


Fig. 6: MAE values achieved by various XGBoost parameter configurations on the balanced test-set. The x-axis encodes the number of samples used for training. The best MAE for each parameter configuration is given in the legend.

D. Comparison to 3D Skeleton Estimation

In order to give an insight on how other approaches perform on our data, we applied two recent SotA Deep Learning 3D skeleton estimation approaches from [14] and [15] on our test set. Even though we are not able to retrain these approaches on our training set, we assume that the large amount of data they were trained with provides good generalization capabilities. To calculate the person's orientation from the estimated 3D skeleton, we used the cross product from the left to right shoulder vector and the vector from the left shoulder to the spine base. Note: For the coco model in [15] the spine base is interpolated from both hip joints. The resulting vector is then projected onto the ground plane and the $acrtan$ function is used to calculate

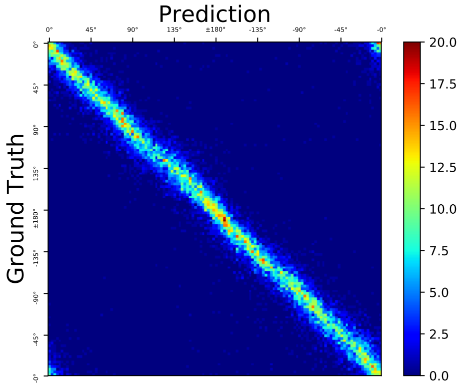


Fig. 7: Confusion matrix of our best XGBoost regression model discretized to 128 bins.

the orientation angle. Results are depicted in Tab. III. It can be seen that our estimations are much more accurate than the ones extracted from [14]. This is reasonable since we use depth as an additional source of information. Orientation estimations results from [15], are slightly better than our fast approach but worse than our precise one. However, when it comes to computation times, the presented pipeline clearly outperforms the deep learning competitors without the need of specialized hardware. In combination with the preferred detector from [21], our fast approach achieves a frame rate of ∼12fps while our precise one runs at ∼6fps, when one person is in the scene.

VI. CONCLUSIONS

We presented a fast and accurate approach for person orientation estimation based on colored point clouds. Our approach achieves real-time estimation rates with low computational costs on a consumer CPU and is, therefore, particularly suitable for mobile robotic applications. We compared the common orientation estimation methods of multi-class classification and regression on a novel data set. Moreover, we have shown that the former approach for attribute estimation [18] is also able to work in the continuous domain. Hence, we expect it will also work for other real-valued person attributes, like age or weight.

REFERENCES

[1] P. Papadakis, A. Spalanzani, and C. Laugier, "Social mapping of human-populated environments by implicit function learning," in *International Conference on Intelligent Robots and Systems (IROS)*, Nov 2013, pp. 1701–1706.

[2] T. Amaoka, H. Laga, S. Saito, and M. Nakajima, "Personal space modeling for human-computer interaction," in *Entertainment Computing (ICEC)*, 2009, pp. 60–72.

[3] X. Truong and T. Ngo, "to approach humans?: A unified framework for approaching pose prediction and socially aware robot navigation," *IEEE Transactions on Cognitive and Developmental Systems (TCDS)*, vol. 10, no. 3, pp. 557–572, 2018.

[4] E. Avrunin and R. Simmons, "Using human approach paths to improve social navigation," in *International Conference on Human-Robot Interaction (HRI)*, 2013, pp. 73–74.

[5] P. Papadakis and P. Rives, "Binding human spatial interactions with mapping for enhanced mobility in dynamic environments," *Autonomous Robots*, vol. 41, no. 5, pp. 1047–1059, 2017.

[6] B. Lewandowski, D. Seichter, T. Wengefeld, L. Pfennig, H. Drumm, and H.-M. Gross, "Deep Orientation: Fast and Robust Upper Body Orientation Estimation for Mobile Robotic Applications," in *will be published on International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[7] Ch. Weinrich, Ch. Vollmer, and H.-M. Gross, "Estimation of Human Upper Body Orientation for Mobile Robotics using an SVM Decision Tree on Monocular Images," in *International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2147–2152.

[8] L. Fitte-Duval, A. A. Mekonnen, and F. Lerasle, "Upper body detection and feature set evaluation for body pose classification," in *VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications*, vol. 2, 2015, pp. 439–446.

[9] F. Shinmura, D. Deguchi, I. Ide, H. Murase, and H. Fujiyoshi, "Estimation of Human Orientation using Coaxial RGB-Depth Images." in *VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications*, 2015, pp. 113–120.

[10] J. Choi, B.-J. Lee, and B.-T. Zhang, "Human body orientation estimation using convolutional neural network," *arXiv preprint arXiv:1609.01984*, 2016.

[11] L. Beyer, A. Hermans, and B. Leibe, "Biternion Nets: Continuous Head Pose Regression from Discrete Training Labels," in *German Conference on Pattern Recognition (GCPR)*, 2015, pp. 157–168.

[12] Y. Kohari, J. Miura, and S. Oishi, "CNN-based Human Body Orientation Estimation for Robotic Attendant," *Workshop on Robot Perception of Humans*, 2018.

[13] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1302–1310.

[14] D. Tome, C. Russell, and L. Agapito, "Lifting From the Deep: Convolutional 3D Pose Estimation From a Single Image," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, "3D Human Pose Estimation in RGBD Images for Robotic Task Learning," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[16] H.-M. Gross, S. Meyer, R. Stricker, A. Scheidig, M. Eisenbach, St. Mueller, Th. Q. Trinh, T. Wengefeld, A. Bley, Ch. Martin, and Ch. Fricke, "Mobile Robot Companion for Walking Training of Stroke Patients in Clinical Post-stroke Rehabilitation," in *International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1028–1035.

[17] H.-M. Gross, St. Mueller, Ch. Schroeter, M. Volkhardt, A. Scheidig, K. Debes, K. Richter, and N. Doering, "Robot Companion for Domestic Health Assistance: Implementation, Test and Case Study under Everyday Conditions in Private Apartments," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5992–5999.

[18] T. Linder and K. O. Arras, "Real-time full-body human attribute classification in RGB-D using a tessellation boosting approach," in *International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1335–1341.

[19] E. Einhorn, T. Langner, R. Stricker, Ch. Martin, and H.-M. Gross, "MIRA – middleware for robotic applications," in *International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2591–2598.

[20] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5636–5643.

[21] B. Lewandowski, J. Liebner, T. Wengefeld, and H.-M. Gross, "A Fast and Robust 3D Person Detector and Posture Estimator for Mobile Robotic Applications," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4869–4875.

[22] L. Spinello, M. Luber, and K. O. Arras, "Tracking people in 3D using a bottom-up top-down detector," in *International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1304–1310.

[23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[24] Y. Kawanishi, D. Deguchi, I. Ide, H. Murase, and H. Fujiyoshi, "Misclassification tolerable learning for robust pedestrian orientation classification," in *International Conference on Pattern Recognition (ICPR)*, 2016, pp. 486–491.

[25] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[26] P. Sudowe, H. Spitzer, and B. Leibe, "Person Attribute Recognition with a Jointly-trained Holistic CNN Model," in *International Conference on Computer Vision Workshop (ICCVW)*, 2015.

[27] W. Liu, Y. Zhang, S. Tang, J. Tang, R. Hong, and J. Li, "Accurate estimation of human body orientation from RGB-D sensors," *Transactions on Cybernetics*, vol. 43, no. 5, pp. 1442–1452, 2013.

[28] Advanced Realtime Tracking GmbH. (2018) ART DTrack2. [Online]. Available: https://ar-tracking.com/

[29] K. Pentenrieder, P. Meier, G. Klinker, *et al.*, "Analysis of tracking accuracy for single-camera square-marker-based tracking," in *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI Fachgruppe VR/AR, Koblenz, Germany*, 2006.

[30] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[31] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.