in: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012), Vilamoura, Portugal, pp. 2147-2152, IEEE 2012

Estimation of Human Upper Body Orientation for Mobile Robotics using an SVM Decision Tree on Monocular Images

Christoph Weinrich, Christian Vollmer, Horst-Michael Gross

Abstract— **In this paper, we present a monocular, texture-based method for person detection and upper-body orientation classification. We build on a commonly used approach for person recognition that uses a Support Vector Machine (SVM) on Histograms of Oriented Gradients (HOG) [1] but replace the SVM by a decision tree with SVMs as binary decision makers. Thereby, in addition to the pure detection of persons, the distinction of eight upper-body orientation classes is enabled. The detection of humans and the estimation of their upper-body orientation from larger distances is essential for socially acceptable navigation of mobile robots. It permits to estimate the human's notice of the robot or even the human's interest in an interaction. Thus, it is the basis for the decision whether to approach or to avoid a human. By using an SVM decision tree for upper-body orientation estimation in discrete steps of 45°, we were able to classify about 64% of the test samples with an absolute error of less than 22.5°. This performance is much better than the results we obtained with comparable methods. Furthermore, our approach proved to be faster than the other state-of-the-art methods. This is of high relevance for implementation on mobile robots with limited computational resources.**

I. INTRODUCTION

The recognition of human upper-body orientations is an important requirement to improve human-robot interaction (HRI), for example in the field of socially acceptable navigation. Especially in public and sometimes busy environments, like supermarkets or home improvement stores [2], the navigation is part of nonverbal communication and has socio-emotional importance. To optimize the HRI, it is necessary that the robot's navigation behavior is socially acceptable for the users of the robot and for uninvolved bystanders. Particularly, assistant or guiding robots need to be articulate, kind, and non-intrusive. A socially acceptable navigation behavior of such robots is substantially influenced by the spatial relation between the robot and its surrounding persons [3]. Thus, it is the basis for the decision whether to approach or to avoid a human. Furthermore, these approach-or-avoid behaviors themselves should respect the human's personal space [4] and therefore be adapted to the human's position and upper body orientation [5]. Thereby, a special challenge is to detect persons and estimate their upper-body orientation, using the limited on board computing power.

In this paper, we investigate the application of a decision tree, where at each node a binary decision is made by a linear SVM and the final classification is determined by the leaves of the tree. The schematic depiction in Fig. 1 shows the separation of three classes. Actually, one background-only class and 8 upper-body classes are separated (Fig. 2). We call those *goal classes*. A texture-based approach, the Histograms of Oriented Gradients (HOG), is used as feature descriptor. However, other feature descriptors, like Linear Binary Patterns (LBP) proved suitable as well and will be investigated further in future research.

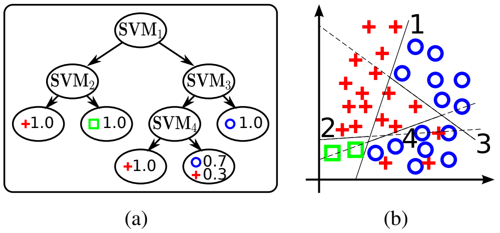


Fig. 1. (a) Logical illustration of Support Vector Machines (SVMs) with tree architecture for exemplary classification of three classes (cross, square, circle) and (b) the geometric depiction of the separation in 2D feature space

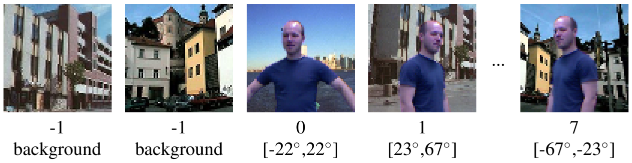


Fig. 2. Training samples with their class labels and the according upper-body orientation domain

After this introduction, Sec. II reviews related work. Afterwards, the used data and the acquisition thereof is described in Sec. III. In Sec. IV our approach for constructing and training an SVM decision tree is described in detail. Sec. V shows that our approach has superior performance w.r.t. related approaches.

II. RELATED WORK

Many approaches for estimating the human upper-body orientation depend on multiple cameras [6], [7], [8] or laser range finders [9], [10] to perceive different views of the persons. Thus, such approaches are not applicable on a mobile robot. Approaches that are based on active depth sensors [11], [12], like Kinect$^{\text{TM}}$, are less applicable as well, as they are limited due to the required data bandwidth, computational and power resources, interferences through external IR light sources, that are common in our scenario, and the maximum distance between 5 and 10m. Approaches that perform on monocular camera images, like the detectors in [1], [13], [14], [15], would be advantageous, because they could be used with wide-angle cameras or even omni-directional cam-

This work has received funding from the Ph.D. Graduate School on Image Processing and Image Interpretation at Ilmenau University of Technology.

C. Weinrich, C. Vollmer, and H.-M. Gross are with Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, 98694 Ilmenau, Germany christoph.weinrich at tu-ilmenau.de

eras. These cameras are relatively inexpensive, have high information content and are nowadays standard equipment on many mobile robots. Furthermore, all people in the robot's environment can be perceived and potentially recognized up to great distance, so that the robot has enough time to react during its tour.

The feature representations obtained by Histograms of Oriented Gradients (HOG) [1] and Local Binary Patterns (LBP) [13] are very robust to changes of color or illumination, making them well suited for our approach, as well.

In [1] and [13] linear SVMs are used for classification. SVMs generalize very well for linear separable problems, and their application is computationally relatively inexpensive, since the classification effort is reducible to one scalar product of the D-dimensional feature vector x and the trained weight vector w, plus a comparison with a threshold b:

$$f(x) = \begin{cases} 1 & \text{if } \sum_{d=1}^{D} w_d \cdot x_d > b \\ -1 & \text{else} \end{cases} \tag{1}$$

Another advantage of linear SVMs is that the basic training parameters consist of only a cost factor, which defines a trade-off between generalization and accuracy, and a tolerance of a termination criterion. Optionally, the cost factor of particular classes or even individual data samples might be multiplied by some weight.

In this work, we distinguish nine goal classes (a background-only class and eight orientation classes). For classification with SVMs, this problem has to be mapped to multiple binary-class problems, for which several methods exist that will be explained in the following.

A one-versus-rest (1-v-r) multi-class SVM [16] employs C binary SVMs to separate each of the C classes from the remaining classes. The resulting classification is done by a winner-takes-all strategy, whereas the classifier with the highest output value determines the class.

The one-versus-one (1-v-1) method [17] applies $\frac{C^2-C}{2}$ binary SVMs to separate C classes from each of the remaining classes. A max-wins voting strategy is used, where the vote for a class is increased whenever this class wins the binary classification. For robust classification, each class should be separable by a binary SVM from each of the remaining classes. This is less constricting compared to 1-v-r SVMs, because the separation of one class against all others is generally harder.

Directed Acyclic Graph SVMs (DAGSVMs) [18] employ $\frac{C^2-C}{2}$ binary SVMs to separate C classes, as well. But in contrast to the 1-v-1 method, these binary SVMs are hierarchically arranged. Thereby only $C-1$ binary classifications have to be accomplished to classify one sample.

If the above mentioned multi-class methods are applied to linear SVMs, each separation between a pair of classes (1-v-1) or between one class and all other classer (1-v-r) is done by one hyperplane. This decreases the robustness of the classifier on data that is not perfectly linearly separable. In the experiments, we will show that linear multi-class SVMs like 1-v-1 and in particular 1-v-r are not able to separate the orientation classes properly. One obvious solution would be the use of nonlinear SVMs. However, these are computationally more expensive, because all k support vectors have to be processed, and for each of them the kernel function $K(x^{(i)},x)$ needs to be computed:

$$f(x) = \begin{cases} 1 & \text{if } \sum_{i=0}^{k} \alpha_i K(x^{(i)},x) > b \\ -1 & \text{else} \end{cases} \tag{2}$$

Another disadvantage compared to linear SVMs is that the kernel parameters have to be defined. For example, the widely used Gaussian kernel needs to be parameterized by its covariance matrix. Since the kernel parameters strongly affect the performance, this is not a trivial choice.

Moreover, non-linear SVMs require a great amount of memory during training. Therefore, in [19] non-linear SVMs are combined within a decision tree. This approach is called DTSVM. The decision tree is primarily used to decompose the feature space until it gets more efficiently manageable by the non-linear SVMs, which are located in the decision tree's leaves. So, in contrast to pure decision trees, this decomposition does not need to be performed until the subspace contains only samples of a single class, which often causes over-fitting.

But since linear SVMs already show relatively good results for person detection on HOG data [1], the computational effort of non-linear SVMs during application phase seems disproportional.

Obviously, there is a vast amount of multi-class classifiers with different degree of separability and computational effort, but we concentrate on a multi-class extension to SVMs that increases computational cost only minimally. Thereto, we investigated different approaches of SVM decision trees, where at the nodes SVMs are used as binary decision makers. An important distinguishing feature of these trees is the way the multiple classes are mapped on two classes for each binary decision. In [20], e.g., the classes with the closest centers are merged iteratively until only two classes are left. Like the typical decision tree algorithms, the training data is used to train a node, then the training data is separated according to the decision in that node, and the subsets are used to recursively train the child-nodes. Compared to pure decision trees like in [21], the separation of the feature space is not limited to be par-axial and therefore, potentially, generalizes better. As Fig. 1 shows, decision trees can separate concave distributions of data samples and solve multi-class problems. Moreover, the path length from the root node to the leaves reflects the separation complexity of the associated subspaces of the feature space. This means, less complex classification problems need less computing time.

Our approach can be used as a detector, and its output can be utilized in later processing stages, e.g. in a person tracker, and merged with outputs of other detectors that use other cues to detect persons, in 3D space. For example, we use this approach to support a silhouette-based approach for continuous estimation of orientation, which suffers from silhouette ambiguities. To support this merging, it is beneficial to have a probabilistic notion of the outputs. Instead of providing just the most likely orientation class, our approach determines the probability of each orientation class. Since the classifier is applied on a multi-resolution pyramid like in [1], the classification results are stored within a multi-resolution pyramid as well, where each level stores the probability of all eight orientation classes. Knowing the camera parameters, a given pose hypothesis in 3D space can be mapped onto a position in the multi-resolution pyramid, yielding the corresponding probabilities or vice versa.

The next section describes the data acquisition of the training and test images of humans standing or moving in different orientations in front of a mobile robot and their ground truth labeling. In Sec. IV the specification of the training of an SVM decision tree, which is especially designed for our application, is shown.

III. Data Acquisition

The training data set consists of about 1.5 million image sections. Each of them is labeled with -1 if it shows exclusively background. The other images that show humans, are labeled from 0 to 7, whereas the label describes the upper body orientation from -180° to 180° in 45° intervals (Fig. 2). The vertical position of the hip is located at the images' lower border and the head is located with a distance that is 10% of the image height, to the upper edge. The human's spine is always located around the center of the image and all images have the same aspect ratio. Certain variations of the image positions are included, because when the sliding window technique is applied to get the detection windows during application phase, the detection windows do not hit each human exactly, either. However, by this relatively strict positioning of the image sections, we make sure that particular body parts are located at almost the same positions, to simplify the classification problem. Furthermore, the detection of humans becomes spatially more precise. The background of each image showing a human is also contained in one of the images that show exclusively background. This prevents the classifier from modeling the goal classes based on the background regions of images that show humans. To increase the variance of the background and reduce the correlation between persons and background, each image that shows a human has another background.

To automatically obtain orientation labels for the training data, which consists of 16 differently dressed people, we used the skeleton tracking facility of the OpenNI$^{\text{TM}}$ together with the Kinect$^{\text{TM}}$ depth camera. The training data was captured of humans which had less than 5m distance to the camera. However, due to the use of a multi-resolution pyramid this does not limit the maximum distance during application of this approach. The skeletons' shoulder positions were used to compute the orientation and to automatically label the images. Furthermore, background subtraction in HSI color space was used to roughly segment the person in the image. Afterwards GrabCut [22] was applied for pixel accurate segmentation of the human. Thereby, the actual background could be replaced by the image segments from the background images (Fig. 2) of the INRIA person data set [1] ("blue boxing"). The resulting data set, which consist of 23,876 images showing humans in different upper-body orientations and about 1.5 million background image segments, was split into a training and a validation data set, at a ratio of 10 to 1.

A test data set of 1,490 images was created with an additional person. In contrast to the training data set, in this case the background of the test images was not replaced. Furthermore, the orientation labels of the test data set were manually checked and if necessary corrected as the orientation estimations delivered by the OpenNI as ground truth data sometimes were incorrect. Additionally, 96,146 background images for the test set were generated from the Caltech background data base [23].

IV. Training of the SVM Decision Tree

For classification, the root node of the decision tree employs a linear SVM to decide which of its child nodes is capable to classify a given data sample. The child nodes recursively do the same until a leave node is reached. As will be explained later, each leave holds a multinomial distribution of labels for those samples that are placed in the subspace that the respective leaf is responsible for. The output of our classification procedure is the most likely class label (Fig. 1) or optionally the whole distribution over the nine class labels.

The decision tree is constructed recursively starting at the root node. To train the SVM of the current node, all training samples in the set S have to be assigned to one of the binary classes -1 and 1, which we call *grouping*. Thereafter, the training data set is separated according to the SVM decision, and for each subset S^{-1} and S^{1}, which are the subsets of all samples with binary label -1 and 1, respectively, this algorithm is applied recursively to the child nodes. The recursion is terminated when a certain proportion $p \in (0,1]$ of a node's training samples S belong to one class, or further separation does not improve the classification on the validation data set, which indicates over-fitting. Increasing p generally leads to a reduction of the resulting tree's depth. The distribution of a node's training samples over the orientation labels is estimated by the relative frequency of the training samples that 'fell' in the responsibility of this node.

To estimate the quality of an SVM node w.r.t. to the accuracy of the whole tree, the information gain $G(SVM)$ is used on the validation data set. It defines the reduction of the entropy within the child nodes $H(S^{-1})$ and $H(S^{1})$ compared to the entropy of the parent node $H(S)$. The information gain is also applied to grow decision trees by the ID3 algorithm [21]:

$$G(SVM) = H(S) - \frac{|S^{-1}|}{|S|} H(S^{-1}) - \frac{|S^{1}|}{|S|} H(S^{1}) \tag{3}$$

$$H(S) = -\sum_{c=0}^{C} \frac{|S_c|}{|S|} \log \frac{|S_c|}{|S|} \ , \tag{4}$$

where S_c is the set of samples of goal class $c \in \{0, \ldots, C\}$. To tune the SVM's training parameters of each node, w.r.t. the gain G(SVM), we propose the application of five steps (Fig. 3), whereas some are optional. Each step will be described in the following.

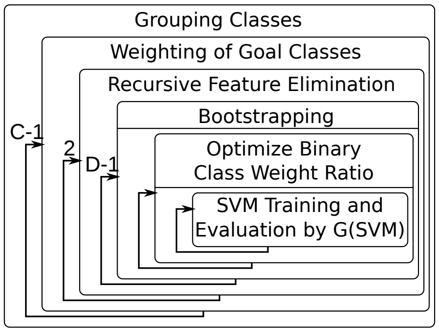

Grouping Classes
Weighting of Goal Classes
Recursive Feature Elimination
Bootstrapping
Optimize Binary Class Weight Ratio
SVM Training and Evaluation by G(SVM)
C-1
2
D-1

Fig. 3. Steps for tuning of the SVM parameters. The steps are applied recursively, e.g., for each of the $C-1$ groupings the weighting is applied

A. *Binary Grouping of Classes*

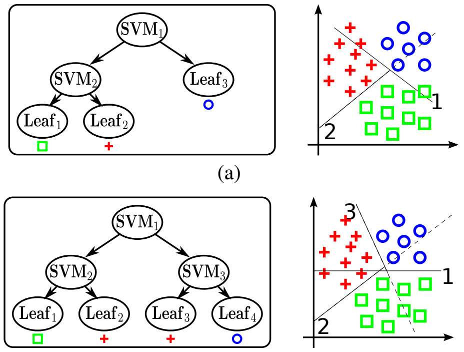
To train a node's SVM each of the training samples has to be assigned to a binary class. For simplification, all samples of one goal class c are mapped to the same binary class $b(c) \in \{-1,1\}$. The resulting task is to find a good mapping m of goal classes to binary classes $m = \{b(1), b(2), ..., b(C)\} \in \{-1,1\}^C$. Initially the goal class with most of the samples $c' = \arg\max_c |S_c|$ is mapped to the binary class 1 and all the other classes are mapped to -1. Then iteratively m is modified by changing the mapping of that goal class $c'' = \arg\max_{\forall c, b(c)=-1} \frac{|S_c \cap S^1|}{|S_c|}$, which has a binary label $b(c) = -1$ and was classified worst according to the old mapping. Thus $C-1$ mappings are tested by actually training an SVM. The best grouping is dependent on the data samples. So, besides two 1-versus-rest separations, $C-3$ separations of two groups of goal classes are checked.

B. *Optional Weighting of Goal Classes*

Ideally, all training samples are perfectly separable w.r.t. the binary labels. If this is not the case, the SVM attempts to separate all training samples as good as possible without any notion of the training samples' goal classes. However, regarding the SVM decision tree, it is more important to separate individual goal classes as good as possible, even if the overall classification gets worse. Accordingly, it shall be avoided that the training samples of certain goal classes negatively affect the separation of other goal classes. Therefore, the goal classes of each binary group, that have a greater classification error than the best classified class, are temporarily deleted (or down-weighted) from the training data set and a second SVM is trained on this data. Thereafter, it is checked whether this SVM reaches a better information gain $G(SVM)$ on the validation data than the initially trained SVM.

C. *Optional Recursive Feature Elimination (RFE)*

To prevent the trained SVMs from over-specialization, the irrelevant features should be eliminated from the training data set. An easy way is greedy backward feature selection for linear SVMs with RFE [24]. Thereby, all features are ranked by their reduction of the SVM's margin. Then, some features with low rank are eliminated, and the SVM is trained again. More features might be eliminated iteratively. For a linear SVM, the rank of feature d is given directly by the trained weight vector w: $rank_d = |w_d|$. Since the information gain $G(SVM)$ is calculated using the validation data set and the SVM is trained on the training data set, finally these features are selected where the SVM shows best generalization abilities. This way for different nodes different features might become relevant.



(a)

(b)

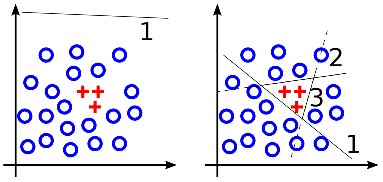
Fig. 4. The effect of weight reduction of that class, which is illustrated by +, is shown in (b). Compared to (a) the margin of SVM_1 between □ and ○ increases and generalization improves. However, in return one more SVM is needed to separate + from ○

D. *Bootstrapping*

The training time of SVMs is dependent on the number of training samples. Some of the training samples have very little influence on the trained SVM. In order to prevent that these samples increase the training effort unnecessarily, bootstrapping is applied iteratively. Initially a suboptimal SVM is trained on a random subset of the training samples. This SVM is used to classify further random samples and falsely classified samples are added to the training set. This is repeated until each sample of the training set is classified correctly, or it is already included in the training set.

E. *Optimization of Class Weight Ratio*

When an SVM is trained, its operating point on the ROC curve is dependent on the weight ratio for the two classes. By default this ratio is 1.0. In this step, the ratio is adapted by exponential binary search as long the FPR = 1.0 or the FNR = 1.0. This is relevant, when the training data is not linearly separable and, e.g., an enclosed class is represented by relatively few samples (Fig. 5).



(a)

(b)

Fig. 5. SVM trained on two not linearly separable classes, which are equally weighted 5(a). In 5(b) the class, which is pictured as red crosses is weighted up. Therewith, the separation 1 is achieved

V. Experiments

Before presenting the results of the SVM decision tree for upper-body orientation estimation, the classification quality and computational effort will be compared to some approaches that have been mentioned in Sec. II. To that purpose, the benchmark data sets that have also been used in [19] are employed. These data sets of different application areas can be downloaded from the UCI Machine Learning Repository [25]. Additionally, the HOG Upper-Body (HOG-UB) data set is created from the acquired images (Sec. III), whereas an HOG descriptor with 72x72 pixel window size, 16x16 pixel block size, and nine gradient orientation bins per cell is applied. This results in a 2,304-dimensional feature vector for each training image. A characterization of all data sets is shown in table I. For each data set the number of classes, the feature dimension and the number of training, validation and test samples are listed.

TABLE I
Evaluation data sets from [19] and our HOG-UB

	#class	#dim	#training	#validation	#testing	#sample
PHW	10	16	7,227	1,872	1,893	10,992
Letter	26	16	13,294	3,336	3,370	20,000
Shuttle	7	9	38,664	9,573	9,763	58,000
Poker	10	10	16,674	4,165	4,171	25,010
CI	2	14	30,148	7,537	7,537	45,222
Forest	7	54	387,343	96,835	96,834	581,012
PPI	2	14	836,544	206,635	206,635	1,249,814
KDD	5	41	3,265,623	816,405	816,403	4,898,431
HOG-UB	9	2304	1,411,880	141,185	70,636	1,623,701

For the smaller data sets, the accuracy (ACC) is shown in table II (where *SVM Tree* refers to our approach), and the number of CPU cycles that are required to classify all test samples are listed in table III. It is shown that the accuracy of our SVM tree is higher than the one of the pure decision tree and the pure linear multi-class SVM. Furthermore, the computational effort is similar to these approaches. The classification rate of our approach is lower than the one of DTSVM, but it needs only a fraction of the computation time.

TABLE II
Accuracy[%] on smaller test data sets

	PHW	Letter	Shuttle	Poker	CI
DTSVM[19]	99.52	97.66	99.89	56.75	84.81
SVM Tree	98.42	89.14	99.98	53.97	84.22
ID3 Tree[21]	95.51	87.18	99.95	49.72	80.97
1-vs-rest SVM	91.86	70.77	91.23	49.94	83.29

TABLE III
CPU cycles for classification of smaller test data sets

	PHW	Letter	Shuttle	Poker	CI
DTSVM[19]	$1.4 \cdot 10^{8}$	$7.6 \cdot 10^{9}$	$1.8 \cdot 10^{7}$	$1.0 \cdot 10^{9}$	$7.0 \cdot 10^{8}$
SVM Tree	$2.7 \cdot 10^{6}$	$5.6 \cdot 10^{6}$	$5.4 \cdot 10^{6}$	$7.3 \cdot 10^{6}$	$1.3 \cdot 10^{7}$
ID3 Tree[21]	$2.4 \cdot 10^{6}$	$5.1 \cdot 10^{6}$	$5.5 \cdot 10^{6}$	$9.5 \cdot 10^{6}$	$1.8 \cdot 10^{7}$
1-vs-rest SVM	$3.6 \cdot 10^{6}$	$8.8 \cdot 10^{6}$	$1.7 \cdot 10^{7}$	$7.7 \cdot 10^{7}$	$1.3 \cdot 10^{7}$

Table IV and V show the accuracy, or the balanced accuracy (BAC) respectively, and computational effort of the larger data sets. The HOG-UB data set is strongly unbalanced in favor to the background class (-1). Therefore, the BAC is used for evaluation instead of the accuracy.

TABLE IV
Classification quality on large test data sets

	Forest ACC[%]	PPI ACC[%]	KDD ACC[%]	**HOG-UB** BAC[%]
DTSVM[19]	94.59	92.29	99.99	51.61*
SVM Tree	94.41	90.99	99.99	64.82
ID3 Tree[21]	93.31	88.11	99.99	33.07
1-vs-rest SVM	71.50	87.42	99.81	55.23

TABLE V
CPU cycles for classification of large test data sets

	Forest	PPI	KDD	**HOG-UB**
DTSVM[19]	$4.2 \cdot 10^{9}$	$3.0 \cdot 10^{10}$	$2.3 \cdot 10^{9}$	$1.3 \cdot 10^{11}$
SVM Tree	$6.8 \cdot 10^{8}$	$1.9 \cdot 10^{9}$	$1.5 \cdot 10^{9}$	$2.6 \cdot 10^{9}$
ID3 Tree[21]	$6.5 \cdot 10^{8}$	$1.9 \cdot 10^{9}$	$1.9 \cdot 10^{9}$	$1.0 \cdot 10^{10}$
1-vs-rest SVM	$2.6 \cdot 10^{8}$	$3.1 \cdot 10^{8}$	$2.5 \cdot 10^{9}$	$8.8 \cdot 10^{9}$

Especially on the high dimensional HOG-UB data set the application of linear SVMs, like in 1-vs-rest SVM and SVM Tree, perform very well. Both approaches have relatively low computational cost and good classification accuracy. For classification of the nine goal classes, the SVM Tree outperforms all tested classifiers w.r.t. accuracy and computational effort. Unfortunately, the DTSVM implementation could not process the complete HOG-UB data set on our 3.5 GHz training PC with 128 GB RAM, because of the storage needs. Thats why the DTSVM was trained on 10% of the HOG-UB data set. This took 45 hours, whereas our approach took 1.4 hours for training on the complete data set.

The error between estimated orientation and actual continuous ground truth orientation is presented as histogram in Fig. 6. It shows that about 64% of the test samples were classified with an absolute error below 22.5° which is half the orientation range of one class.

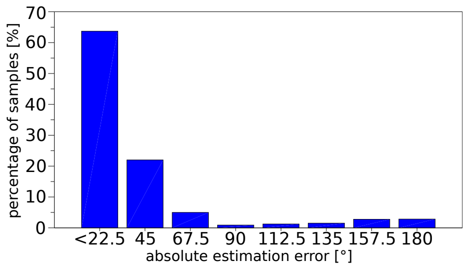


Fig. 6. Histogram of the samples' proportion over the absolute classification error between estimated, discrete orientation and ground truth orientation

The true positive rate (TPR) for pure person detection regardless of the upper-body orientation is 97.7% and the false positive rate (FPR) is 0.87%. An actual classification result of the detector is visualized in Fig. 7. Note, that we did not group the bounding boxes resulting from the different levels of the scale space like commonly done by e.g. the mean shift algorithm.

The topology of the resulting decision tree is shown in Fig. 8. Due to the binary grouping of the goal classes (see section IV-A), the class samples, that are very close in feature space, are separated by nodes further down the decision tree. The

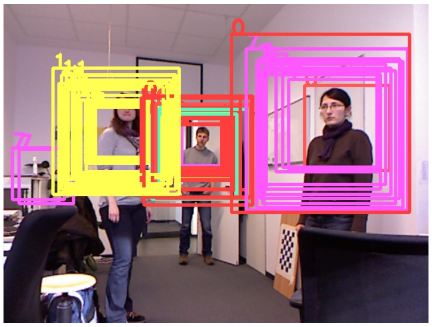


Fig. 7. Example image of person detection with upper-body orientation: Yellow stands for for the 45°, red for the 0° and magenta for the -45° class. Only bounding boxes (BB) with an output probability of over 0.65 are shown for visualization purposes. Since we did not apply the usual grouping of the BBs, the BBs of all levels of the scaling pyramid are shown

deepest non-leaf nodes show that neighboring orientation classes, as well as class 4 (averted) and 0 (frontal) with similar silhouette are most difficult to separate. The depth of the decision tree is only five, and the background class, which is presented to the classifier most often, is ideally separated already after two decisions. This is the reason why our decision tree is three times faster than the 1-versus-rest multi-class SVM, which always applies 9 binary SVMs.

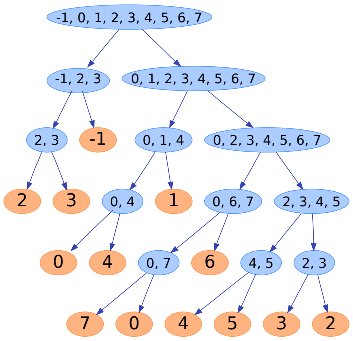


Fig. 8. The SVM decision tree which was trained on our HOG-UB data set. The numbers within the blue nodes show all goal classes where the node was trained on. The numbers within the red leaves show the most likely class of the leave

Thus, the computational cost for HOG feature extraction and upper-body orientation classification on images of size 640 × 480 increases on average by only 21% from 682ms to 824ms on a 2.8 GHz PC compared to the use of one binary, linear SVM for pure detection.

VI. Summary and Conclusions

We have shown that an SVM decision tree for classification of HOG descriptors can be used to successfully detect humans and to classify their upper-body orientation. It is important to note that the computation time for the feature extraction and classification only increased by 21% compared to the use of a single linear SVM for pure detection. Thus the proposed approach is still applicable on mobile robots. We plan to fuse the output of our method together with other cues within a probabilistic tracking framework [26] and will use the orientations to improve several socially acceptable navigation behaviors on our socially assistive robot [2].

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, 2005, pp. 886–893.

[2] Gross et al., "TOOMAS: Interactive shopping guide robots in everyday use - final implementation and experiences from long-term field trials," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009, pp. 2005–2012.

[3] M. Svenstrup, S. Tranberg, H. Andersen, and T. Bak, "Pose estimation and adaptive robot behaviour for human-robot interaction," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 3571–3576.

[4] E. T. Hall, "A system for the notation of proxemic behavior," *American Anthropologist*, vol. 65, no. 5, pp. 1003–1026, 1963.

[5] Satake et al., "How to approach humans?: Strategies for social robots to initiate interaction," in *Proc. Conf. on Human Robot Interaction (HRI)*, 2009, pp. 109–116.

[6] L. Rybok, M. Voit, H. Ekenel, and R. Stiefelhagen, "Multi-view based estimation of human upper-body orientation," in *Proc. 20th Int. Conf. on Pattern Recognition (ICPR)*, 2010, pp. 1558 –1561.

[7] M. Hofmann and D. Gavrila, "Multi-view 3d human pose estimation in complex environment," *Int. Journal of Computer Vision (IJCV)*, vol. 96, no. 1, pp. 103–124, 2012.

[8] M. Voit and R. Stiefelhagen, "A system for probabilistic joint 3d head tracking and pose estimation in low-resolution, multi-view environments," in *Proc. of the 7th Int. Conf. on Computer Vision Systems (ICVS)*, 2009, pp. 415–424.

[9] T. Kanda, D. Glas, M. Shiomi, and N. Hagita, "Abstracting peoples trajectories for social robots to proactively approach customers," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1382 –1396, dec. 2009.

[10] Glas et al., "Laser tracking of human body motion using adaptive shape modeling," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007, pp. 602–608.

[11] Shotton et al., "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. on Comp. Vision and Pattern Rec.*, 2011, pp. 1297–1304.

[12] D. Droeschel and S. Behnke, "3d body pose estimation using an adaptive person model for articulated icp," in *Proc. 4th Int. Conf. on Intelligent Robotics and Applications (ICIRA)*, 2011, pp. 157–167.

[13] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *Proc. IEEE 12th Int. Conf. on Computer Vision (ICCV)*, 2009, pp. 32 –39.

[14] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[15] J. Wu, C. Geyer, and J. M. Rehg, "Real-time human detection using contour cues," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[16] V. N. Vapnik, *Statistical learning theory*, 1st ed. Wiley, Sept. 1998.

[17] U. H.-G. Kreßel, *Pairwise classification and support vector machines*. Cambridge, MA, USA: MIT Press, 1999, pp. 255–268.

[18] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems (NIPS)*, 2000, pp. 547–553.

[19] F. Chang, C.-Y. Guo, X.-R. Lin, and C.-J. Lu, "Tree decomposition for large-scale svm problems," *Journal of Machine Learning Research (JMLR)*, vol. 11, pp. 2935–2972, 2010.

[20] H. Osman, "Novel multiclass svm-based binary decision tree classifier," in *Proc. IEEE Symp. on Signal Proc. and Information Techn. (ISSPIT)*, 2007, pp. 880–883.

[21] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[22] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transact. on Graphics*, vol. 23, pp. 309–314, 2004.

[23] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, 2003, pp. 264–271.

[24] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, pp. 389–422, 2002.

[25] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[26] S. Müller, E. Schaffernicht, A. Scheidig, H.-J. Böhme, and H.-M. Gross, "Are you still following me?" in *Proc. 3rd European Conference on Mobile Robots (ECMR)*, 2007, pp. 211–216.