Pixelwise Object Class Segmentation based on Synthetic Data using an Optimized Training Strategy

Frank Dittrich, Heinz Woern
Institute for Process Control and Robotics
Karlsruher Institute of Technology (KIT)
Karlsruhe, Germany
Email: {frank.dittrich, woern}@kit.edu

Vivek Sharma, Sule Yayilgan
Faculty of Computer Science and Media Technology
Gjovik University College
Gjovik, Norway
Email: {vivek.sharma, sule.yayilgan}@hig.no

Abstract—**In this paper we present an approach for low-level body part segmentation based on RGB-D data. The RGB-D sensor is thereby placed at the ceiling and observes a shared workspace for human-robot collaboration in the industrial domain. The pixelwise information about certain body parts of the human worker is used by a cognitive system for the optimization of interaction and collaboration processes. In this context, for rational decision making and planning, the pixelwise predictions must be reliable despite the high variability of the appearance of the human worker. In our approach we treat the problem as a pixelwise classification task, where we train a random decision forest classifier on the information contained in depth frames produced by a synthetic representation of the human body and the ceiling sensor, in a virtual environment. As shown in similar approaches, the samples used for training need to cover a broad spectrum of the geometrical characteristics of the human, and possible transformations of the body in the scene. In order to reduce the number of training samples and the complexity of the classifier training, we therefore apply an elaborated and coupled strategy for randomized training data sampling and feature extraction. This allows us to reduce the training set size and training time, by decreasing the dimensionality of the sampling parameter space. In order to keep the creation of synthetic training samples and real-world ground truth data simple, we use a highly reduced virtual representation of the human body, in combination with KINECT skeleton tracking data from a calibrated multi-sensor setup. The optimized training and simplified sample creation allows us to deploy standard hardware for the realization of the presented approach, while yielding a reliable segmentation in real-time, and high performance scores in the evaluation.**

I. Introduction

The application of the here proposed approach for pixelwise segmentation of human body parts in RGB-D sensor data is intended in research scenarios related to safe human-robot cooperation (SHRC) and interaction (SHRI) in the industrial domain. In our experimental environment we allow for a shared workspace with no spatial and temporal separation between human worker and industrial-grade components and robots. In the context of SHRC and SHRI, we focus on the intuitive and natural human-robot interaction, safety considerations and measures in a shared work environment, the realization of cooperative processes and the workflow optimization.

All elements of our research spectrum thereby rely on information related to activities in the workspace. As a basis for the information generation on different levels of abstraction, we use a multi-sensor setup which delivers RGB-D data in a high frequency. This sensor data is then further processed by low-level image processing approaches for optical flow estimation or pixel-wise object class segmentation, by mid-level approaches for object class detection and human body posture recognition and by high-level approaches for action recognition and situation awareness. The results from the different approaches are thereby interchanged, and the hierarchical scene analysis represents the core of a modular cognitive system for safe human-robot collaboration.

In the here presented approach we directly process the depth measurements of a RGB-D sensor which is placed on the ceiling in the center of the shared workspace, in order to provide detailed and spatially resolved information about distinct body parts in the scene in real-time. This information then serves the scene analysis modules for inference and planing on higher abstraction levels.

The remainder of this paper is organized as follows. In Section II, related work concerning object class segmentation is presented. In Section III we describe our approach in detail. In Section IV, the performance of our approach is evaluated and discussed. Finally, in Section V, a conclusion is drawn and hints for future work are given.

II. Related Work

The segmentation of 2D Data from visual sensors is a complex problem in low-level image processing and many approaches have been proposed over the years. Applications in domestic and industrial robotics, autonomous driving cars and internet search optimization often build up on the semantic analysis of RGB and depth images, where approaches in this field in turn often rely on segmentation information about object classes contained in the images. Most segmentation approaches thereby provide a labeling of each pixel, where a label depicts the affiliation to a certain object or the background class.

The application of Probabilistic Graphical Models (PGM), especially Conditional Random Fields (CRF) for the labeling problem is one of the major techniques for finding an optimal image labeling or segmentation ([6], [12], [7], [9]). These models allow for a convenient way to statistically model interactions of distinct information sources for the optimization process. In [6], He et al. used this property to incorporate segmentation information on different pixel patch scales, with a single pixel on the lowest scale. Here, several filters were used to predict label information of the patches on different scales. In [12], Yao et al. also use this property in order to perform holistic scene understanding, where information on different semantic abstraction levels is used for the joint reasoning about the pixel labeling, the location of objects and the scene type. Here, a TextonBoost approach ([9]) delivers information on the pixel level. The main disadvantage of those models is the complexity of the optimization process, which in most cases is not tractable because of the high number of pixel nodes in the graphical representation, and their modelled interactions. In [7], Krähenbühl and Koltun present an efficient inference technique, which allows for a fully connected CRF, where the pixelwise interactions have to be modeled by a linear combination of Gaussian kernels. The fully connected CRF model applied to a standard object class segmentation task showed promising results, while real-time conditions for inference could be met.

In [9], Shotton et al. mention that most of the performance of their segmentation approach is based on the information of the pixelwise classification, and the modeled pairwise pixel interactions in a 4-neighborhood only serve the regularization or the so called filling in effect in the optimization process, which mostly results in smoother and *prettier* results. It is therefore, that many researchers abandon the CRF modeling and focus on the pixelwise classification, without considering the label context. In [9], Shotton et al. use a boosted classifier for this task.

Lepetit and Fua ([8]) where one of the first who used a Random Decision Forest (RDF) classifier for a low-level classification task in image processing. In their publication they showed how object recognition based on local features can be efficiently done by shifting most of the computation time for keypoint recognition into the training of a classifier. In this context they demonstrated the high performance and the low training complexity of RDFs, because of the randomness in the classifier training. Also they used synthetic data for the classifier training, which was generated by applying affine transformations to real image representations of the objects.

Since then, many pixelwise object class segmentation approaches were developed based on RDFs, with different pixel feature descriptions and weak learner types ([11], [3], [4], [5]). In [11], Stückler et al. use a combination of depth and RGB patches centered at pixels as feature descriptions for training and prediction. For the decisions in the nodes of the trees they apply simple difference tests on the normalized sums of random feature sub-spaces. In [3], Dumont et al. use RGB patches centered at pixel, as feature description for training and prediction. In their approach, the weak learner type simply uses threshold tests of random dimensions of the feature space. For the training of the RDF, a maximum randomization concept is applied. Also, in parts of their approach, the label context of the whole pixel patch is considered in the training procedure. Kontschieder et al. ([4]) present in their work the use of the label context for training and prediction, for the segmentation of 2D RGB images. The use of the label context in the prediction step, showed more coherent segmentation results, comparable to labeling results from CRF based approaches with simple 4-neighborhood pairwise potentials. In [5], Shotton et al. demonstrate the application of human body part segmentation as a basis for human pose recognition. In their approach, pixel centered patches in the depth data from a RGB-D sensor are used for feature descriptions in training and prediction. All training data was thereby synthetically generated by applying motion caption data to detailed and articulated 3D human body models in a virtual environment. Finally, in [1], a comprehensive survey of RDF based applications in image processing is given, with examples for image segmentation.

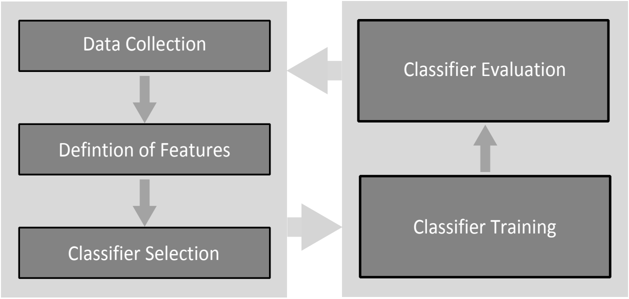

Fig. 1. Schematic layout of a generic classification system. Design decisions determine the training and testing data generation, define the feature space and the corresponding feature extraction, and select the classifier type. Based on these decisions, classifier training and evaluation is conducted. Depending on the evaluation results, the design might be adapted for performance optimization.

III. Classification System for Pixelwise Segmentation

In our approach, we want to perform segmentation of aligned RGB and depth data from a RGB-D ceiling sensor, where each pixel is labeled corresponding to its object class affiliation. For this, we transform the problem into a standard classification task, which allows for the use of a generic system modeling framework (Fig.1). Because of our basic design decisions of the system components, our approach can be compared to the work from [5]. Our object classes are the distinct human body parts: *Head*, *Upper Body*, *Upper* and *Lower Arm*, *Hand*, *Legs* and the background rejection. For the generation of data for the classifier training, we use a synthetic representation of the human body in a virtual environment, where synthetic sensors generate depth data. The features used for the description of the object class samples are based on the depth information only, and are extracted by a centered pixel patch with constant size. Also, we aim for real-time label predictions on images with a resolution of 640×480 pixels. In our approach, for the realization of a reliable and robust segmentation in real-time, we strive for simplicity of the data generation and a time efficient classifier training. This allows for the use of standard hardware components and very short training periods, which renders the realization of our approach simple and feasible. In the following sections we will describe the components of our approach in detail.

A. Data Collection

For the training and evaluation of the classifier, we must generate a large number of pixelwise body part class samples with a known labeling. In case of the training we use solely synthetic depth data, and in case of the testing data for the evaluation step, we use synthetic and real-world depth data. In Section IV we show that synthetic data is sufficient for the generalization of the classifier in regard to real-world data.

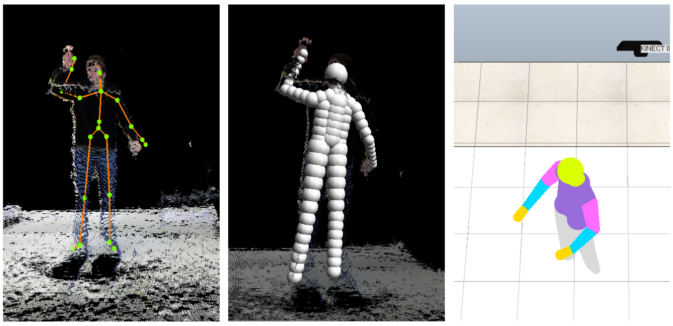

Fig. 2. *Left*: Results from the KINECT skeleton tracking. *Center*: Coarse approximation of the human body, modeled by a small set of spheres arranged along the skeleton estimate. *Right*: Finer sphere approximation of the human body, modeled by a larger set of spheres in the V-REP environment. Notice the synthetic KINECT sensor representation above the human body model.

1) Synthetic Data Generation: The high variability of the appearance of the human body demands for a high number of training samples. In our case the variations concern the body proportions, the body posture, and the translation and orientation of the body relative to the sensor. The appearance variations related to clothing or skin and hair color can be omitted because of the exclusive use of depth data. In our approach, we randomly pose a synthetic representation of the human body in the field of vision of a static synthetic KINECT sensor in a virtual environment. To address the problem of high variation in the first two categories, body proportion and posture, we use a broad spectrum of motion capture data based on the KINECT human body posture tracking (Fig. 2 *left*). Here, persons with differing body proportions perform distinct choreographies with the arms and upper body, facing a real-world KINECT sensor. The result of this procedure is an ordered set of skeleton tracking data:

$$
\begin{aligned}
D_{skel} &= \{T_1, T_2, \ldots\} \ , \\
T_p &= \{C_1, C_2, \ldots\} \ , \\
C_c &= \{S_1, S_2, \ldots\} \ , \\
S_t &= \{\mathbf{a}_1, \mathbf{a}_2, \ldots\} \ \ , \qquad (1)
\end{aligned}
$$

where T_p stands for the entire tracking data of person p, which in turn consists of the different choreographies C_c of types c. A choreography is defined by the estimated skeleton setups S_t at time steps t, where each setup consists of the joint positions $\mathbf{a}_i$. Based on this tracking data, many examples of body proportions and postures can be injected into the training data generation process.

For the synthetic ground truth data generation, we use the virtual robot experimentation platform V-REP (see [2]). This framework allows for a remote access on parts of its functionality via a C/C++ API, and synthetic KINECT sensors are already included. Also, the full version of the software is free for educational and academic use. Here, we create a human body approximation based on a set of spheres (Fig.2 *center*, *right*) for the synthetic representation of the human body in the virtual scene. The spheres are colored according to their object class or respectively body part affiliation (Fig.2 *right*), and positioned distinctively along the various bones and joints of the current skeleton setup. To address the third variation component, the body transformation, we statically position a synthetic KINECT sensor in the virtual scene, corresponding to the position of the real-world sensor, and transform the body representation in its' current setup in 3 dimensions in the floor plane:

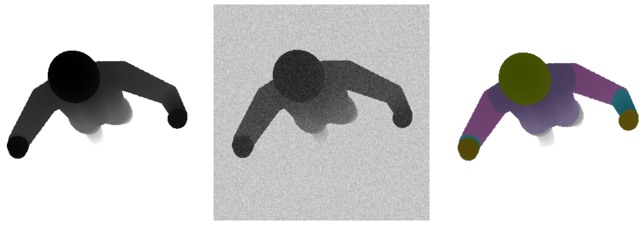

Fig. 3. *Left*: Synthetic depth data generated with a synthetic KINECT sensor and a parameterized human body representation in the field of vision of the sensor. *Center*: Synthetic depth frame with additive white Gaussian noise. *Right*: Overlay of the object class or respectively body part coloring and the synthetic depth data.

$$
\begin{aligned}
T_{floor} &= \mathrm{T_o}\left(\Delta_x, \Delta_y, \alpha\right) \ , \\
\tilde{\mathbf{a}}_i &= T_{floor} \cdot \left(\mathbf{a}_{i_x}, \mathbf{a}_{i_y}, \mathbf{a}_{i_z}, 1\right)^T \ , \qquad (2)
\end{aligned}
$$

where the operator T_o generates a transformation matrix from the 2d translation (Δ_x, Δ_y) in the floor plane, and the rotation α around the floor normal in (Δ_x, Δ_y). All skeleton joints $\mathbf{a}_i$ are then transformed accordingly.

To create a basis for the generation of synthetic training data, we now sample in each step uniformly vectors $\lambda = (p, c, t, \Delta_x, \Delta_y, \alpha)$ from the parameter space, and set up the human body representation in the scene accordingly. Using the functionality of the V-REP framework, we can then retrieve synthetic depth frames from the virtual KINECT sensor in the scene (Fig.3 *left*). Because of the noise in the real-world data, and to cope with *unseen* data samples in the testing step more robustly, meaning to further the generalization ability of the trained classifier, we add additive Gaussian white noise to the depth values (Fig.3 *center*). For supervised training and evaluation of the classifier, we also need to know the actual labeling of the samples. To get this information we overlay the RGB and the depth channels of the synthetic sensor, where the distinct object class coloring of the representation automatically assigns the labels to the depth data (Fig.3 *right*). The result of this procedure is the basis for the pixelwise extraction of synthetic ground truth data, based on human body appearances with high variation in all three categories.

2) Real-world Data Generation: For the evaluation of the classifier, we also want to use real-world data, which means that we also need labeled depth frames of human bodies in the field of vision of the real-world ceiling sensor. One approach would be, to let different people perform distinct choreographies under the sensor, and annotate the recorded depth frames by hand. But this would be a very time consuming and tedious task. Instead, we deploy an automatic annotation approach, where the labeling is inferred from the single spheres of the human body sphere representation in the point cloud.

In our experimental environment we use a multi KINECT sensor setup for scene analysis. All sensors are thereby

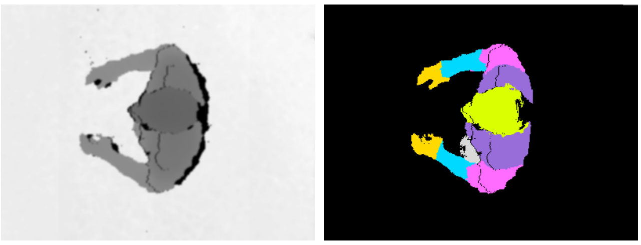

Fig. 4. *Left*: Depth frame from the real-world ceiling sensor, with a person standing in its' field of vision. *Right*: Inferred labeling of the depth data, based on point cloud distance heuristics and the sphere representation of the human body.

calibrated and the mutual transformations are known. It is therefore directly possible to record tracking data from a sensor observing the scene from the side, and depth data from the ceiling sensor (Fig.4 *left*) concurrently, where all data is in one temporal and spatial context. Because of the known transformations, we can transform the skeletal tracking data into the coordinate system of the ceiling sensor. Based on the parameterized sphere representation (Fig.2 *center*) and the given assignment of the spheres to distinct body part classes, we can use distance heuristics to infer the labeling of all depth values in the frame. One automated labeling result is shown in the right image in Figure 4. Although the quality of this example is very high, not all predictions exhibit the same quality. It is therefore necessary, to discard predictions of low quality in a subsequent step by hand, which can be done very quickly. The result of this two step procedure is the basis for the pixelwise extraction of real-world ground truth data.

B. Definition of Features

Based on the data, created as described in the Sections III-A1 and III-A2, we generate our training and test samples. For this we must define the features which are used for the description and classification of the samples. In our approach, we use a rectangular region, centered at the pixel sample position (Fig.5 *left*), for the extraction of depth values around the sample. The ordered depth values are then used as the feature description $\mathbf{f}$ of the object class sample s:

$$\begin{aligned} \mathbf{f}(s) &= \left(f_{[1:w_p],1}, f_{[1:w_p],2}, \ldots, f_{[1:w_p],h_p}\right) \in \mathbb{R}^{w_p \cdot h_p}, \\ f_{i,j} &= \mathrm{d_o}\left(s_x + (i - w_p/2), s_y + (j - h_p/2)\right) \ , \\ & \qquad (i,j) \in \{1,\ldots,w_p\} \times \{1,\ldots,h_p\}, \end{aligned} \tag{3}$$

where (s_x, s_y) is the position of sample s in the depth frame and $\mathrm{d_o}(i,j)$ depicts the operator which returns the depth value at position (i,j) in the depth frame. The values w_p and h_p are the static width and height of the feature region.

In the training step, the classifier therefore learns to discriminate body part classes based on the spatial and geometrical local layout of the samples. The size of the layout region is fixed for all samples, and the region is in accordance to Eq.3 parallel to the coordinate system of the image sensor. Elements of the feature region which are outside the image boundaries are treated as background, which in our case is represented by the depth values of the workspace floor.

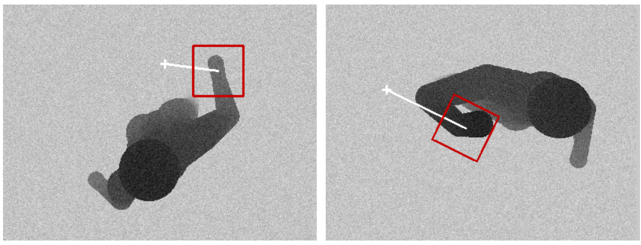

Fig. 5. Feature extraction of a hand pixel sample using a rectangular region. *Left*: The rectangular region is parallel to the image coordinate system and centered at the sample position. *Right*: The rectangular region is centered at the sample and rotated according to the adaptation approach in Section III-C.

Based on the definition of the features and the extraction method, we then generate the training and evaluation data by uniformly selecting annotated depth frames (Sec.III-A1, III-A2) and extracting the features and the label of random body part class samples. The number of randomly chosen frames and samples per class are thereby design parameters for the classifier training and evaluation.

C. Optimized Training Strategy

In our approach, in order to reduce the complexity of the classifier training, we optimized parts of the described processes in Section III-A and III-B. The optimizations thereby only concern the generation of training data, and are independent of the specific classifier.

First, instead of using different persons with varying body proportions for the skeletal choreography recording, we only use one person with average proportions. To inject variation in regard to the body proportion into the synthetic depth frame generation, we apply simple scaling of the recorded skeleton setups:

$$S_{scaled} = \lambda \cdot S_{orig} = \{\lambda \cdot \mathbf{a}_1, \lambda \cdot \mathbf{a}_2, \ldots\} \quad . \tag{4}$$

At creation time, instead of uniformly sampling the person parameter p, we sample scaling factors λ from a fixed interval $[\lambda_{min}, \lambda_{max}]$. This measure simplifies the generation of training data with high variation in body proportions, but it does not reduce the required amount of training samples for a high classifier performance.

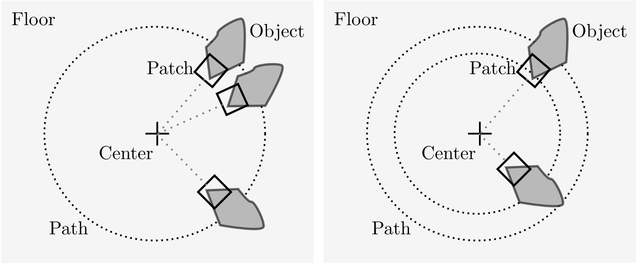

Fig. 6. Feature patch adaptation. *Left*: A patch layout centered at an object point with adapted orientation according to the object point and center position, which is invariant if the object moves on a circular path around the center point. *Right*: Changing distance causes slow changes in the adapted path layout.

In order to reduce the number of required training samples, we apply an elaborated and coupled strategy which is based on the following assumptions. Imagine a sensor placed on the ceiling with the image sensor parallel to the floor, and an object is moving on a circular path around the image center projection point on the floor, while always facing the floor normal in the center point with the same object surface (Fig.6 *left*). Then, a pixel patch at a fixed object point, adapted to the object position as depicted in Fig.6, would always have the same layout, independent of the exact position of the object on this path. If the radius of this circular path is changed, the layout of the patch also changes because of the position of the sensor (Fig.6 *right*). The *rate of change* as a function of the distance is however very low, if sensor and object point height are no too far apart.

Based on these assumptions, we adapt the synthetic depth frame generation process and the feature extraction. Instead of uniformly sampling 2D translations (Δ_x, Δ_y), we sample distances d from a small set $D = \{d_1, d_2, \ldots, d_N\}$, where N can be very small because of the assumed low rate of change. The translation parameters are then calculated as:

$$(\Delta_x, \Delta_y) = (c_x + d, c_y + d) \ , \tag{5}$$

where (c_x, c_y) is the projection of the image center on the floor. The result is that during data creation, various skeleton setups are presented to the synthetic sensor in different orientations and on different positions along a straight line. This means, that one dimension of the sample parameter space could be removed. In addition, the variation of d is very small, because of the small set options D. Altogether, these measures reduce the number of required training samples for an adapted feature extraction, because of the elimination of redundancy.

To actually use the reduced synthetic data for training, we also have to adapt the feature extraction as illustrated in Fig.6 and Fig.5 *right*. Accordingly Eq.3 is changed to:

$$\begin{aligned}
\tilde{\mathbf{f}}(s) &= \left(\tilde{f}_{[1:w_p],1}, \tilde{f}_{[1:w_p],2}, \ldots, \tilde{f}_{[1:w_p],h_p}\right) \in \mathbb{R}^{w_p \cdot h_p} \ , \\
\tilde{f}_{i,j} &= \mathrm{d_o}(\mathrm{t}(i,j)) \ , (i,j) \in \{1, \ldots, w_p\} \times \{1, \ldots, h_p\}, \\
\mathrm{t}(i,j) &= (\mathbf{b}_0, \mathbf{b}_1) \cdot \begin{pmatrix} i - w_p/2 \\ j - h_p/2 \end{pmatrix} + \begin{pmatrix} s_x \\ s_y \end{pmatrix} \ ,
\end{aligned} \tag{6}$$

where the function t transforms the patch position (i, j) into a global frame position, using the basis vectors $\mathbf{b}_0$ and $\mathbf{b}_1$ of the rotated region coordinate system. The first basis vector $\mathbf{b}_0$ is thereby defined by the displacement of the pixel sample s relative to the depth frame center $(w_f/2, h_f/2)$. The second basis vector $\mathbf{b}_1$ is defined by the orthogonality constraint.

Both, the reduced synthetic depth frame generation and the adapted feature extraction can now be used for an optimized classifier training, where the classification performance can be preserved, while simultaneously reducing the number of training samples. It should be mentioned, that the adapted feature extraction is also used for testing and the label predictions.

D. Classifier Selection

The choice of the actual classifier is independent of the descriptions in the preceding sections, because of the generic structure of the whole classifier system. In our approach for pixelwise object class labeling, we use Random Decision Forests for the classification task. RDFs have many advantages over other classification methods. These are mainly the ability to perform multi-class classification without any extensions, the fast training and the high generalization ability because of the randomization in the training step, the easy implementation because of the simple structure, the direct possibility of parallelization, the fact that the predictions can be understood as empirical distributions conditioned on the test sample and finally the high classification performance. We will give a very short overview over the principle of RDF training and testing, in order to motivate the different parameters and to describe the weak learner type, which is the basis for the trained decisions in the nodes of the trees. A comprehensive description of RDFs and applications can be found in [1].

A binary Decision Forest F consists of an ensemble of n_t binary Decision Trees $T = \{t_i\}$. A tree t_i has corresponding to its' name a directed binary tree as a graph representation, with two types of nodes: split nodes which exhibit two child nodes and leaf nodes with no child nodes. Split node represent decisions based on distinct trained feature functions, which are of the same type for all split nodes and trees. Leaf nodes represent the class prediction of a tree. In order to predict a class label of sample s, the sample is routed through the tree according to the decisions of the node feature functions, which process the samples' feature vector $\mathbf{f}(s)$. The leaf node, the sample ends up in, then delivers the prediction for the class label.

When training a tree, a set of training samples with known labels are passed down the tree. In each node the training procedure tries to find the optimal feature function, where optimality considerations are based on quality measures like the entropy. Here, the difference between the entropy of the class label set of the samples at the current node and the mean entropy of the label sets in the child nodes after applying the binary decision to the node samples, serves the evaluation of a certain feature function. If the maximum difference is smaller than a certain threshold, or the maximum tree depth $d_{t_{max}}$ is reached, then the split node becomes a leaf node, and the relative frequency of the different class labels of the training samples assigned to this node are used for the estimation of the empirical class distribution, which in turn is used for the class label prediction.

Training of a forest is done by training the single trees on all training samples, and for testing the empirical class distributions of all trees are used for a forest prediction. For instance the maximum mean class probability can be used as the forest prediction for a test sample.

Random Decision Forests are Decision Forests where randomness is injected into the training process, in order to speed up the training and to further the generalization ability and robustness of the classifier. This can be done by randomly choosing subsets of the training samples for the single tree training (bagging), or by randomly choosing fixed sized subsets of feature space dimensions for the decisions in the slit nodes. In our approach we use both techniques. For the bagging we apply training data sampling with replacement, and for the decisions based on a random feature subspace we use a linear discrimination of 2D subspaces with thresholding of the

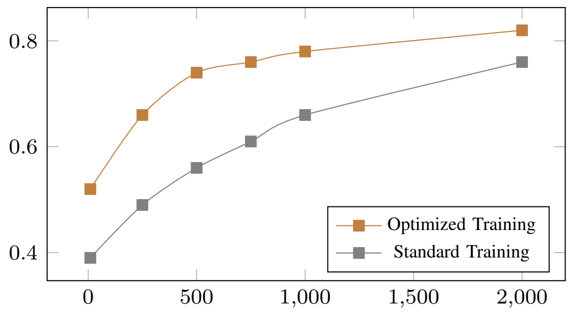

Fig. 7. Comparison of the standard and optimized training strategy, using the average Recall measure (vertical axis) as a function of the number of synthetic depth frames used for training (horizontal axis). For the evaluation, 250 annotated real-world depth frames were used.

distance to the linear discrimination border:

$$(f_{d_1}, f_{d_2})(b_1, b_2)^T \geq \delta \ . \tag{7}$$

Therefore, our feature functions are parameterized by the choice of the two feature dimensions $\{d_1, d_2\}$, the orientation (b_1, b_2) of the linear discrimination and the distance threshold δ. For the training we can control the randomness by constraining the number of randomly sampled feature function parameters for the optimization in each node.

IV. Evaluation

For the evaluation of the overall segmentation approach and the optimized training, we use a fixed parameter setup with forest size $n_t = 5$, feature patch size $(w_p, h_p) = (64, 64)$ and maximum tree depth $d_{t_{max}} = 15$. For the randomization in the training process we use 100 threshold and 100 feature function samples in the node optimizations, and bagging with replacement for the tree-wise training data sampling. All training is based on synthetic depth frames with additive white Gaussian noise using a standard deviation of 15 cm. For the performance evaluation we use the Recall and Precision measure for single object classes and the average as the combined measure for all classes ([10]).

The numbers presented in Table I - III, and the prediction results illustrated in Fig.8 are based on the same trained decision forest. Here, a total of 5000 synthetic depth frames, generated as described in III-C, were used as a basis for the optimized RDF classifier training. For the training process of each tree, 2000 frames from this data were chosen randomly, and for each frame, 300 pixel positions per object class were chosen uniformly for the extraction of the features patches and ground truth labels. Altogether, this resulted in approximately 2.6×10^6 synthetic labeled training samples per tree, with a training time for the whole forest of approximately 40 min using a PC with Intel i7 CPU and 4 GByte RAM. Calculating the pixelwise predictions for a frame with 640×480 pixels, using the trained forest, takes about 40 ms on this hardware.

TABLE I. Confusion Matrix using Synthetic Data

		Bg	He	UB	UA	LA	Ha	L
Bg	(Background)	**0.95**	0.00	0.00	0.00	0.00	0.00	0.05
He	(Head)	0.00	**0.93**	0.05	0.01	0.01	0.00	0.00
UB	(Upper Body)	0.00	0.03	**0.87**	0.08	0.00	0.00	0.02
UA	(Upper Arm)	0.00	0.00	0.16	**0.80**	0.04	0.00	0.00
LA	(Lower Arm)	0.00	0.00	0.02	0.14	**0.78**	0.06	0.00
Ha	(Hand)	0.00	0.00	0.00	0.02	0.23	**0.75**	0.00
L	(Legs)	0.00	0.00	0.04	0.00	0.01	0.00	**0.95**

TABLE II. Confusion Matrix using Real-World Data

	Bg	He	UB	UA	LA	Ha	L
Bg	**0.95**	0.00	0.00	0.00	0.00	0.00	0.05
He	0.00	**0.84**	0.08	0.02	0.05	0.01	0.00
UB	0.00	0.00	**0.83**	0.15	0.02	0.00	0.00
UA	0.00	0.00	0.19	**0.67**	0.13	0.01	0.00
LA	0.00	0.00	0.00	0.05	**0.77**	0.18	0.00
Ha	0.00	0.00	0.00	0.04	0.15	**0.81**	0.00
L	0.03	0.00	0.04	0.02	0.01	0.03	**0.87**

When applied to synthetic and real-world testing data, the trained RDF produced similar quantitative and qualitative results for both data types, as presented in Table I - III and Fig.8 respectively. Overall, the testing of the synthetic data shows better results compared to the real-world data, yet the quantitative measures are not far apart and demonstrate a good overall performance for both types. This indicates, that the training concept based on synthetic data only, using a coarse approximation of the human body in limited postures and transformations is sufficient for the reliable and high-performance segmentation of real-world data, in our application scenario.

In order to demonstrate the usefulness of our optimized training strategy, we evaluated trained forests on 250 annotated real-world depth frames, using the standard and optimized strategy, and a varying number of synthetic samples for training. The results depicted in Fig.7 thereby show a steeper ascent and higher average Recall values for our optimized strategy. When compared, the quality measures of both meet at 1000 and 250 training samples for the standard and the optimized strategy respectively. This indicates the ability to learn the highly varying appearance of object classes based on a reduced number of training samples in case of the optimized training.

V. Conclusion

In this paper we described a generic classification approach for the pixelwise labeling of object classes, applied to the problem of human body part segmentation in RGB-D data from a ceiling sensor. As an innovation we presented an optimized training strategy which allows for a reduced number of training samples while preserving the classification performance. Also, for the classifier training, we demonstrated the applicability of simple synthetic human body representations in a virtual environment, the use of the KINECT skeleton estimations as a substitute for motion capturing data and the elaborated combination of both for the automated ground truth

TABLE III. Confusion Matrix based Quality Measures

	Avg	Bg	He	UB	UA	LA	Ha	L
Recall_Synth	**0.86**	0.95	0.93	0.86	0.79	0.77	0.75	0.94
Precision_Synth	**0.71**	1.00	0.97	0.79	0.77	0.72	0.63	0.11
Recall_Real	**0.82**	0.94	0.84	0.83	0.67	0.76	0.80	0.87
Precision_Real	**0.61**	1.00	0.99	0.70	0.65	0.48	0.46	0.03

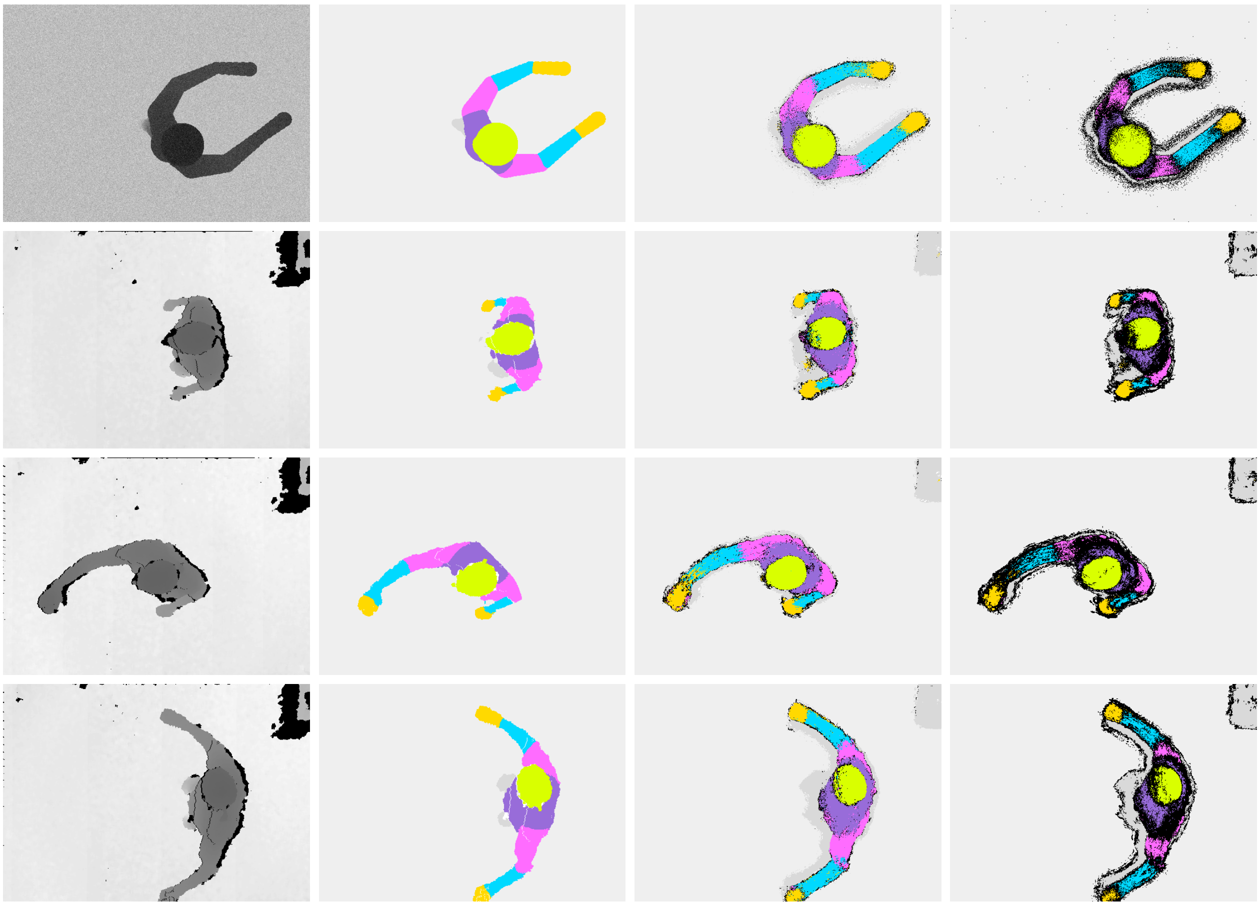

Fig. 8. Prediction results based on synthetic and real-world data. The first column shows the feature frames based on depth data, the second column shows the ground truth labeling, the third and fourth column show the prediction results with prediction probability thresholding of 0.5 and 0.75 respectively. Class predictions with a probability less than the thresholds are colored black in the result images. The first line is based on synthetic testing data, the second to fourth lines are based on real-world testing data.

labeling of real-world data. The quantitative and qualitative results presented in the evaluation thereby emphasize the high performance of the overall system and the suitability of synthetic training data for the segmentation of real-world data.

References

[1] A Criminisi and J Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer Publishing Company, Incorporated, 2013.

[2] M. Freese E. Rohmer, S. P. N. Singh. V-rep: a versatile and scalable robot simulation framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[3] Dumont et al. Fast Multi-class Image Annotation with Random Subwindows and Multiple Output Randomized Trees. In Alpesh Ranchordas and Helder Arajo, editors, *VISAPP (2)*, pages 196–203. INSTICC Press, 2009.

[4] Kontschieder et al. Structured class-labels in random forests for semantic image labelling. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2190–2197, November 2011.

[5] Shotton et al. Real-time Human Pose Recognition in Parts from Single Depth Images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304. IEEE Computer Society, 2011.

[6] Xuming He, Richard S Zemel, and Miguel Á Carreira-perpi nán. Multiscale Conditional Random Fields for Image Labeling. In *CVPR*, pages 695–702, 2004.

[7] Philipp Krahenbuhl and Vladlen Koltun. *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*. 2011.

[8] V Lepetit and P Fua. Keypoint Recognition using Randomized Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2006.

[9] J Shotton, J Winn, C Rother, and A Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *In ECCV*, pages 1–15, 2006.

[10] Marina Sokolova and Guy Lapalme. A Systematic Analysis of Performance Measures for Classification Tasks. *Inf. Process. Manage.*, 45(4):427–437, 2009.

[11] Jörg Stückler, Nenad Biresev, and Sven Behnke. Semantic mapping using object-class segmentation of RGB-D images. In *IROS*, pages 3005–3010. IEEE, 2012.

[12] Jian Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012.