Efficient Real-Time Pixelwise Object Class Labeling for Safe Human-Robot Collaboration in Industrial Domain

Vivek Sharma$^{\diamond\star}$
$^{\diamond}$ESAT-PSI, iMinds,
KU Leuven, Belgium

Frank Dittrich
IAR-IPR,
KIT, Germany

Sule Yildirim-Yayilgan
*Gjøvik University College,
NTNU, Norway

Luc Van Gool$^{\diamond\mp}$
$^{\mp}$CVL, BIWI,
ETH Zürich, Switzerland

Abstract

In this paper, we use a random decision forests (RDF) classifier with a conditional random field (CRF) for pixelwise object class labeling of real-world scenes. Our ultimate goal is to develop an application which will provide safe human-robot collaboration (SHRC) and interaction (SHRI) in industrial domain. Such an application has many aspects to consider and in this work, we particularly focus on minimizing the mislabeling of human and object parts using depth measurements. This aspect will be important in modelling human/robot and object interactions in future work. Our approach is driven by three key objectives namely computational efficiency, robustness, and time efficiency (i.e. real-time). Due to the ultimate goal of reducing the risk of human-robot interventions. Our data set is depth measurements stored in depth maps. The object classes are human body-parts (*head, body, upper-arm, lower-arm, hand, and legs*), table, chair, plant, and storage based on industrial domain. We train an RDF classifier on the depth measurements contained in the depth maps. In this context, the output of random decision forests is a label assigned to each depth measurement. The misclassification of labels assigned to depth measurements is minimized by modeling the labeling problem on a pairwise CRF. The RDF classifier with its CRF extension (optimal predictions obtained using graph cuts extended over RDF predictions) has been evaluated for its performance for pixelwise object class segmentation. The evaluation results show that the CRF extension improves the performance measure by approximately 10.8% in F1-measure over the RDF performance measures.

1 Introduction

In this paper, the proposed approach of pixelwise object classification using depth maps is intended for use in areas of challenging industrial environment (Fraunhofer IFF, 2015) for safe human-robot collaboration (SHRC) and interaction (SHRI). The objective is to apply the resulting system in our robotic workspace for classification of objects in areas of interest of the robot in real-time. High safety standards with full elimination of any kind of possible risk of injury to humans and the optimization of the workflow must be met in industrial workspace. The demonstration of the model incorporates the interaction of humans with different industrial-grade objects while optimizing interaction and collaboration processes. The proposed approach covers a wide range of applications in human-robot interaction: for reliable collision detection, in manufacturing operations hand-in-hand with humans, in aviation/automobile industry for integrating aircraft/automobile components, for efficient handling and logistic tasks for fetch-and-carry services, in health care industry that facilitates minimally-invasive-surgery, in medical and rehabilitation sectors, to control traffic services.

In classification as a segmentation task, minimizing misclassification of labels assigned to each pixel is a research area with a lot of room for further research (Boykov et al., 2001) (Boykov et. al., 2006). In this paper, we focus on RDF and CRF approaches for this purpose. We chose RDF for classification, since RDF can generalize more than support vector machine (SVM). The major advantage of RDF over SVM and boosting is that random forests can handle both binary and multi-class problems with the same classification model.

In our approach, depth measurements are directly processed to provide an accurate and spatially resolved information about the available object classes (human, table, chair, storage, and plant) in the scene in real-time from "*top-view*" . Our work is primarily intended for manufacturing and automation industry, where we keep track of humans and industrial-based objects from "*top-view*" for a proactive task. Meantime, it is important to reduce the

Appearing in Proceedings of the 4^{th} Workshop on Machine Learning for Interactive Systems (MLIS) 2015, Lille, France. JMLR: W&CP volume 43 Copyright 2015 by the authors.

cost of surveillance. It is infeasible in terms of costs to use an expensive set of computers, sensors, and algorithms for segmentation. Therefore, our goal is to obtain an efficient and robust segmentation with computational efficiency and real-time support using minimum hardware and software gadgets. Our main contributions are as follows:

- This work is intended for manufacturing and automation industry in challenging environments for SHRI and SHRC. Our work is a step towards scene analysis, and we are proposing a model for pixelwise object class segmentation formulated as a labeling problem in low-level vision tasks for industrial domain. The proposed approach covers a wide range of applications in human-robot interaction: (a) works in real time, (b) with reduced mislabeling error compared to state-of-the-art, (c) is depth invariant, (d) aims at identifying human-robot occlusions (hence reduces risks of accidents due to robot hitting a human in the same workspace), (e) uses for testing, real world data composed of images taken by scenes from real-world, where there are real world objects and humans, (f) focuses on object-object and human-object occlusions.
- The work is general and our synthetic training adapts well to real-world scenarios with good segmentation results. For demonstration, the resulting integrated system is tested in our robotic workspace for segmentation in real-time using our proposed approach (see row 2-3 of Fig. 11).

The remainder of the paper is structured as follows. In Section 2, the related work is given. Section 3, describes feature selection, RDF, CRF and Energy Minimization techniques. In Section 4, data collection and experimental setup is explained. In Section 5, results, discussion and the experimental evaluation is given and in Section 6, we discuss the conclusion and future work.

2 Related Work

Object class segmentation aims to assign a class label to every pixel in the image. Object class segmentation can be formulated as a CRF based labeling problems (Gonfaus et al., 2010) (He et al., 2004) (Shotton et al., 2009) in which a label is assigned to a pixel corresponding to an object class. In general, an object class labeling problem is formulated as maximum-a-posteriori estimation over a CRF, which is generalized as an Ising-Potts model (Boykov et al., 2006). In such a framework of labeling, one aims to assign a label to a pixel which represents an object class and which minimizes the energy for the most *optimal labeling* (Boykov et al., 2001).

(He et al., 2004) use CRFs to incorporate segmentation information with varying scales of a pixel patch in order to predict label information and model context (i.e. contextual information at the global and local levels) with the help of coarser and more global features. The modeled CRF uses a single pixel on the lowest scale in order to segment and recognize the object class of the pixel. In our case, we also process a single pixel by the low-level image processing for pixelwise object class labeling, finally leading to scene analysis.

(Gonfaus et al., 2010) propose a new framework which is based on CRF and which is able to encode any possible combination of class labels varying from local (pixel and super-pixel), mid-level (feature of neighbouring regions), to global scale (taking into account the entire image). The authors combine context at various scales for joint classification and segmentation. (Shotton et al., 2009) propose a segmentation approach purely based on pixelwise classification using boosted classifier. The authors commit that the performance of their segmentation is efficient because of the information obtained from pixelwise classification. Due to this reason, researchers have stepped out of complex CRF modeling and focused on the pixelwise classification, without considering the label context.

(Lepetit et al., 2006) apply random forests on simple binary tests of image intensity neighbouring the key-points for object recognition. They were the first to apply RDF classifier for a low-level classification task in computer vision. They validated that high performance and low training complexity of RDFs was due to the randomness in the classifier training. Since then, RDFs became popular for pixelwise object class segmentation approaches with different pixel feature descriptions and *weak learner* (or split function) types.

(Shotton et al., 2013) demonstrate the application of segmentation of human body-parts to human pose segmentation in real-time using random forests. In their approach, they trained an ensemble of random decision trees based on a pixel centered patch in the depth data obtained from RGB-D sensor. They accomplished fast and impeccable prediction of human pose in real-time.

In this paper, we propose a generic classification technique for pixelwise object class labeling using random forests and CRF extension. Our approach is driven by three key objectives namely computational efficiency, robustness, and time efficiency (i.e. real-time) for industrial applications, and it further differs from (Shotton et al., 2013) in the following aspects. In (Shotton et al., 2013), all training data were thereby synthetically generated by applying marker based motion capture to the detailed and articulated 3D human body models in a virtual environment. On the other hand, we use a highly optimized virtual representation of the 3D human skeleton modeled on a set of 173 spheres in a virtual environment. We generate the synthetic data of the human body-parts in a virtual environment (Freese et al., 2010), using a multi-sensor KINECT setup for skeleton tracking in real world (Dittrich et al., 2014) (see Fig. 4). We generate both real and sythetic data for objects in addition to humans. This way computational expense is reduced. We use "*top-view*" whereas in (Shotton et al., 2013) also include "*front-view, side-view*". Still, our results are competitive as will be shown in the Section 5 .

Some other work, (Sung et al., 2011) use human poses to understand the human activity and the holistic scene. While (Grabner et al., 2011) propose imaginary poses of human appearance to detect objects in the scene. (Jiang et al., 2012) learn the human activities by inferring object arrangements and interaction in a 3D scene.

3 Proposed System

Fig. 1 shows the schematic layout of the segmentation system. Our approach consists of two phases: a) training of an RDF classifier with synthetic data, and b) testing of the trained classifier with new real-world data. Fig. 1 demonstrates the two phases. Given a collection of data comprising input ground truth image and its corresponding depth map obtained from RGB-D sensor in the synthetic world using VREP simulator (Freese et al., 2010). The first step performed is sampling, i.e. the number of frames and samples/depth-measurements per class are chosen randomly for classifier training and evaluation. Next, individual features $\mathbf{v}(s)$ are extracted from 2D patches corresponding to object classes. Then, selected features are passed to the RDF classifier. RDF returns a trained classification forest. Now a test depth map obtained from the real-world KINECT sensor is given as an input to the trained classification forest. The result obtained is a pixelwise object class labeling. The likelihood of an object label assigned to a depth measurement/pixel is then integrated in a pairwise CRF as a unary term in the CRF energy.

3.1 Features Selection

Vector $\mathbf{v}(s)$ represents features from a 2D patch corresponding to an object class and it contains depth measurements in the 2D patch $(h \times w)$. $h \times w$ is the dimensionality of the 2D patch and is the feature space in our case, where h resembles height and w resembles width of the 2D patch (see Fig. 2). The feature description $\mathbf{v}(s)$ of the object class s is based on the depth information only:

$$\mathbf{v}(s) = (f_{[1:w],1}, f_{[1:w],2}, ..., f_{[1:w],h}) \in \Re^{w.h} \qquad (1)$$

$$f_{i,j} = Op(s_x + (i - \frac{w}{2}), s_y + (j - \frac{h}{2})), \quad (i, j) \in \{1, ..., w\} \times \{1, ..., h\}, \qquad (2)$$

where s is the maximum number of object classes, which include: human body-parts (*head, body, upper-arm, lower-arm, hands, legs*), table, chair, plant and storage. (s_x, s_y) is the position of sample in the depth map, the function $Op(.,.)$ returns the depth value from a given position of the depth map.

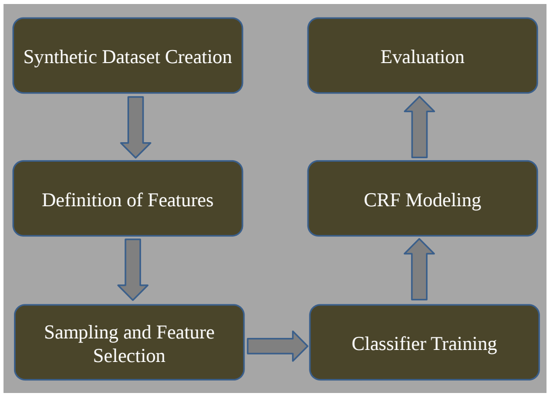

Figure 1: Schematic layout of the segmentation system which shows the steps of data collection, definition of features, sampling and feature selection, classifier training, CRF modeling and evaluation.

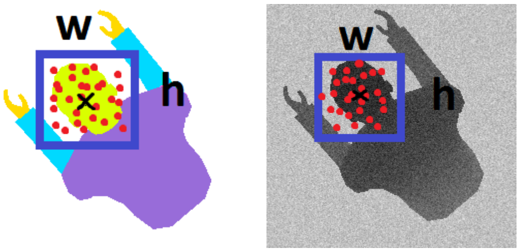

Figure 2: Feature extraction of a head pixel sample using a rectangular region. *Left*: Ground truth coloured-labels of depth data/map. *Right*: Synthetic depth map generated with a synthetic KINECT sensor. In both of the corresponding frames, the rectangular region is parallel to the image coordinate system and centered at the sample position. h and w resembles the height and width of the rectangular 2D patch. The cross sign depicts the center of the patch and the red dots are the randomly chosen features for the tree training.

3.2 RDFs for Object Class Segmentation

We use random decision forest (or random forest) for the classification task. Random decision forests is an ensemble of binary decision trees, which gives an impressive high accuracy on previous unseen data, Such phenomenon is known as generalization. The methods such as bagging and randomized node optimization injects randomness into the trees during the training phase. Bagging helps to avoid specialization of selected parameters to a single training set and improves generalization, while randomized node optimization optimizes each node of the decision tree with respect to the subset of entire parameter space of the weak learner. This approach of producing diversity via randomization has proved to be very efficient and valuable.

In a random forest, the features selection primarily depends on optimization function and on the basis of that the split takes place using a split function. A slight difference in the training sets produces a very high variance in decision tree, randomly selection of features has proved out to improve the prediction results with higher efficiency. Each of the nodes in a decision tree is associated with a split function. In our approach, we employ a linear discrimination corresponding to a single inequality test, where [.] is the $0-1$ indicator. The split function at internal nodes is parameterized by the choice of a simple difference between two feature dimensions $\{d_1, d_2\}$ of the vector $\mathbf{v}$ and thresholded by the distance τ:

$$\left[\mathbf{v}_{d_1} - \mathbf{v}_{d_2} \geq \tau\right] \tag{3}$$

This difference makes the approach depth invariant. At each node, we calculate 100 candidate offset pairs ($\mathbf{v}_{d_1} - \mathbf{v}_{d_2}$) and 100 candidate thresholds τ per offset pairs, i.e. 100×100 comparisons for all split nodes.

The optimization of split function proceeds in a greedy approach. At each node, maximization of information gain is calculated. Based on it, the incoming training samples are split into "best" disjoint subsets of training dataset, i.e. two child nodes are constructed: left-child and right-child node. The procedure is repeated recursively for all the newly constructed child nodes until a stopping criterion for tree growth is met. The stopping criteria in our case were: maximum depth that a tree could reach, the relative frequency of training samples within a leaf node are similar to each other, and unavailability of enough training samples. When any of these stopping criteria is met the split node becomes the leaf node.

After the decision tree is built, each node contains a subset of labeled training samples, then an empirical class distribution is calculated for each leaf node. This is how the binary classification tree is built and an ensemble of more than 2 trees is called a decision forest. Constructing each tree on a different random subset of the training samples (i.e. bagging) or choosing a subset of dimensions at random out of a feature space helps producing diversity and improved generalization by avoiding specialization of selected parameters to a single training set. When a forest is built this way with randomization it is called a random decision forest or random forest.

We know each leaf node of a trained tree represents the class prediction of the tree. Given a new sample (i.e. test sample) $\mathbf{v}'$, it is routed through the trained tree and the goal is to infer the class label c. The testing sample traverses the tree until it reaches a leaf node. At each split node a test is applied and the sample is likely to end up in the leaf with training samples which are similar to itself. At the leaf node the empirical class distribution is read off $P(c|\mathbf{v}')$ and it is reasonable to assume that the sample which ended up in a leaf node must also have an associated label similar to itself. So the label leaf statistics predicts the label associated with the test input sample $\mathbf{v}'$. If there are t trees in a forest, each tree leaf yield the posteriori $P_t(c|\mathbf{v}')$, then the forest class posteriori can be defined as:

$$P(c|\mathbf{v}') = \frac{1}{T}\sum_{t=1}^{T} P_t(c|\mathbf{v}'), \tag{4}$$

where $t \in \{1,...,T\}$ and T is the maximum number of trees in the forest. The forest class posteriori is obtained by averaging each tree posteriori. Fig. 3 shows the graphical approach to above discussed tree testing.

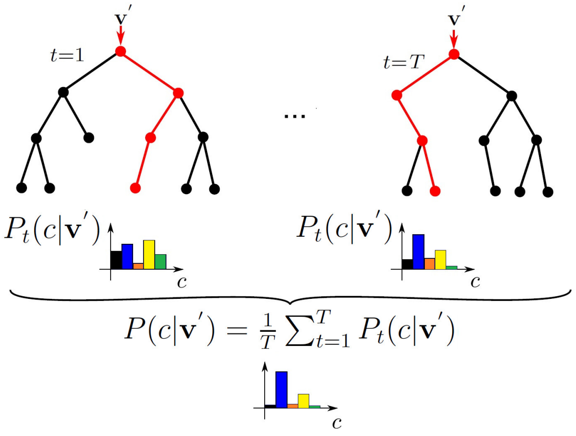

Figure 3: An example of a simple pixelwise object class labeling using RDF classifier: a query test pixel ($\mathbf{v}'$) routes through each trained decision tree in a forest. Each test pixel traverses the tree through several decision nodes until it reaches the leaf node and is assigned a stored leaf statistics of the leaf node $P(c|\mathbf{v}')$, where c is the class label. The forest class posterior is obtained by averaging individual tree posteriors.

3.3 CRFs for Object Class Segmentation

The energy of the pairwise CRFs used for object class segmentation can be defined as a sum of unary and pairwise potential terms as:

$$E(\mathbf{x}) = \sum_{i \in \upsilon} \varphi_i(p_i) + \sum_{i \in \upsilon, j \in \eta} \varphi_{i,j}(p_i, p_j), \tag{5}$$

where υ corresponds to the vertex set of 2D grid with each vertex corresponding to pixel p in the image and η is a neighborhood of the pixels. $\mathbf{x}$ is an arbitrary configuration or labeling. The unary (likelihood) and pairwise (smoothness prior) potential terms of the CRF energy takes the form described next in detail. The unary potential $\varphi_i(p_i)$ term of the CRF energy is the likelihood of an object label assigned to pixel i, obtained from the RDF classifier. The pairwise potential $\varphi_{i,j}(p_i, p_j)$ term (prior term) is of the form of Ising-Potts model (Boykov et al., 2006), which can be efficiently minimized by α-Expansion (move making algorithms). α-Expansion based on graph cuts (Boykov et al., 2001) are used to minimize the pairwise potential smoothness term of the CRF energy function. α-Expansion applies iteratively min-cut/max-flow procedure to an appropriately constructed pairwise CRF and is guaranteed to find an optimal solution, which is close to the global optimal solution. In vision applications, like pixel labeling and segmentation, the best approximation is in the case of graph cuts using α-Expansion with Ising-Potts model, where the final energy is almost equal to the global minimum energy. (Boykov et al., 2001) proved that the minimum energy of the α-Expansion move algorithm in the worst case will be at maximum twice of the global minimum energy. Thus, in simple words the RDF is only trained and the RDF predictions are injected as the data term in the energy formulation of the CRF, and then we do global optimization using the graph cuts algorithm.

4 Data Collection

We generate an extensive dataset for the task of multi-class image segmentation based on industrial domain using Virtual Robot Experimentation Platform (Freese et al., 2010). The data is generated in two phases: a) synthetic training depth data and b) real-world test depth data, with corresponding labeled ground truth data for all object classes. Pixels in the depth map indicate the depth measurement rather than color or intensity measurement of the scene. It is very important and a crucial factor to have a large amount of high precision depth and ground truth data during the training phase for learning a realistic model. Here, the pixelwise RGB-D data has been synthetically generated in a virtual environment for 10 object classes. Using synthetic data for training is very efficient and for that reason, using synthetic data removes the need to annotate the data manually. Our generated pixelwise RGB-D dataset is composed of frames with a "*top-view*" of human body-parts and industrial-grade components. It is publicly available for academic and research purposes.

Synthetic and Real-World Data: Here, we discuss the generation of training and testing human data for synthetic and real-world. We generate the synthetic data of the human body-parts (*head, body, upper-arm, lower-arm, hand, and legs*) with a 3D human model in a virtual environment (Freese et al., 2010), using a multi-sensor KINECT setup for skeleton tracking estimation in real world (see Fig. 4). The generation of human data is based on appearances from an industrial environment with broad spectrum of challenging poses, orientation, shapes, and variable heights. Human appearance includes: *sitting, standing, walking, working, dancing, swinging, boxing, tilting, bending, bowing, and stretching* with combinations of angled arms, single and both arms and other combinations. The human height ranges between 160-190 cm. Due to the high variation in human data, there is a large number of training samples for the classifier training. The more the varied training samples, the better the classifier is trained. Therefore, it is necessary to synthesize a large and varied synthetic training data set using the real-world poses from an industrial environment. We expect that the testing of the generated synthetic data based on real human postures should give better results compared to the real-world data. For real-world data generation, we use a KINECT sensor placed at the ceiling. For more detailed information about human data generation, refer to (Dittrich et al., 2014). In Fig.(5 - 6), we show examples of synthetic human data and real-world human data.

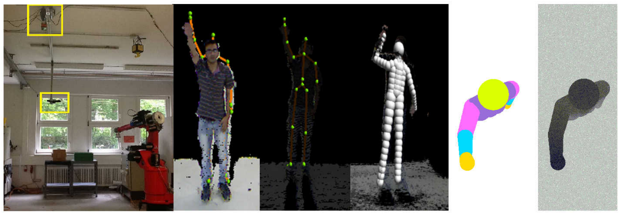

Figure 4: Synthetic human data generation. (*From Left to Right*:) Multi-sensor KINECT skeleton tracking setup at our robotic workplace. Real-world human skeleton tracking (one of our author standing) using KINECT, skeletal joints of interest of real-world human, 3D human skeleton modeled on a set of 173 spheres, ground truth labeling of depth data and corresponding depth data (when KINECT sensor is above the human model at a height of 3.5 meters).

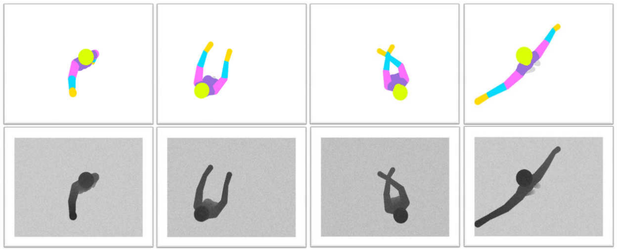

Figure 5: Synthetic human data for training. (*Top*) Ground truth labels of synthetic depth data (*Bottom*).

The generated synthetic data (i.e. depth data and ground truth labels of depth data) for industrial-grade object classes are shown in Fig.(7-8), where Fig. 7 shows the synthetic data for table, chair and Fig. 8 shows the synthetic data for plant, storage.

Dataset Modeling: We also incorporate modeling a 3D scene using multiple 3D objects in a virtual environ-

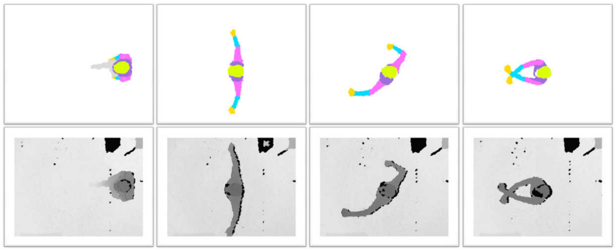

Figure 6: Real world human data for testing. (*Top*) Ground truth labels of synthetic depth data (*Bottom*).

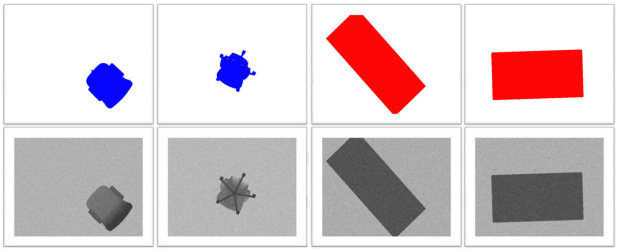

Figure 7: Synthetic data of chair and table for training. (*Top*) Ground truth labels of synthetic depth data (*Bottom*).

ment for obtaining various possible configurations, arrangements, and interactions between human-object and object-object relationships for synthetic dataset generation. This way, the synthetic dataset is made more realistic in comparison to the real-world scenarios. This considerably plays a significant role in the correct and better classification of human body-parts and object parts, while occlusions of human-object or object-object are being recognized. This modeling of our dataset makes our solution usable in identifying occlusion compared to (Shotton et al., 2013) (Dittrich et al., 2014). For the synthetic dataset collection, the workspaces were modeled in the virtual environment to maintain consistency with the real-world targeted workspaces and scenarios, but recognizing the same set of objects. For more detailed information about dataset modeling, refer to (Sharma et al., 2015).

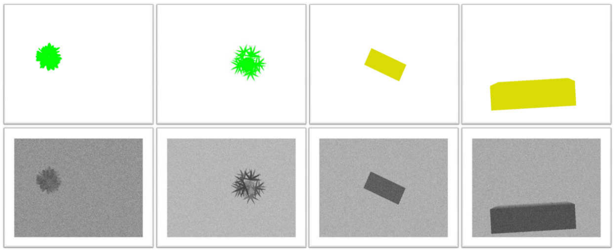

Figure 8: Synthetic data of plant and storage for training. (*Top*) Ground truth labels of synthetic depth data (*Bottom*).

5 Experimental Evaluation, Results and Discussion

To evaluate the overall segmentation approach, we use both synthetic and real-world depth maps. The goal of our work is to do high quality segmentation in real-time and reduce mislabeling errors. For demonstration of our work performance, we discuss the effects of tree depth, number of training frames on the tree training, training time, testing time and finally evaluation of the results obtained form RDF classifier and the CRF extension.

A scene is a single frame where there is a single 3D object or a combination of multiple 3D objects with a particular configuration. We have a single synthetic scene with a single object at a time. All the chosen 3D models of objects are based on industrial workspace and office domain. We synthesize 10,000 frames for human object class, and 4000 frames for each of the industrial grade object classes: chair, table, plant and storage. The synthesized scene has depth map ranging between 0-3.5 m. As mentioned before, the human scenes are composed of a high variation of human poses and shapes The height range of industrial grade object classes is between 70-90 cm. For chair, table, plant and storage object classes, 4 instances were chosen for each object class which are: *executive chair with and without chair handles; conference rectangular table and conference round table; shrubs, flowers and plants within pot; small sized shelves and wardrobes* based on industrial workspace and office domain. Each of the frames are still images, having no temporal or motion information. For the training process of each tree, 1600 frames from this dataset are chosen randomly from each object class. A fixed feature patch size (w, h)=(64,64) was used for the whole training process. Each frame generated from a KINECT camera was of size 640 × 480 pixels.

For the RDF tree training, we use a fixed parameter setup with number of training frames (F)=1600, number of features extracted per object class (PC)=300, number of trees in forest (T)=5, tree depth (D)=19, and weak learner (Feat)=Linear-Feature-Response. For the randomization process, the randomization parameter (Ro=200) during tree training comprises of candidate thresholds (τ=Ro/2) per feature and candidate feature (ψ=Ro/2) function samples in the node optimizations. All trainings are based on training of synthetic dataset with additive white Gaussian noise using a standard deviation of (σ)=15 cm.

For the evaluation process, a desktop with Intel i7-2600K CPU at 3.40GHZ (4 core processor), operating system installed on solid state drive and 4GByte RAM was used. We generate recall-precision metrics for the performance evaluation of each single object classes, mean average of recall (mAR) and mean average of precision (mAP) as the combined average of all classes. For demonstration, we generate qualitative results for both synthetic and real-world data and quantitative results for real-world data only.

5.1 Tree Depth

Of all the training parameters, tree depth (D) plays the most critical and crucial role as it directly impacts the capacity of the RDF classifier. If a very shallow tree is built, the classification might suffer from the problem of under-fitting, where the decision boundaries tend to be very coarse with low-confidence posteriors. On the other hand, if a deep tree is built, the classification might suffer from the problem of over-fitting because it starts coming up with decision boundaries. This over-fitting problem is solved by using multiple trees in a forest, which gives better generalization. It is a very big limitation of this parameter (D) and it is very important that an optimal D is chosen precisely in order to avoid the under-fitting and over-fitting problems. The results in Fig. 9 (Column 1) shows that the recall-precision metrics improves gradually with the increase in tree depth and then starts to saturate around D=17 , and there is much less improvement after D=19.

5.2 Number of Training Frames

The results in Fig. 9 (Column 2) shows that the recall-precision metrics improve gradually with the increase in number of training frames, and then the trained tree starts to saturate around F=1600, and there is much less improvement after F=1600. We found that the increase in number of training frames monotonically increases the testing prediction only if the training set is highly varied (i.e. redundancy of training samples do not let the decision forest learn more, but the precision gains more at the expense of recall). Fig. 10 shows the pixelwise prediction results based on the same trained decision forest.

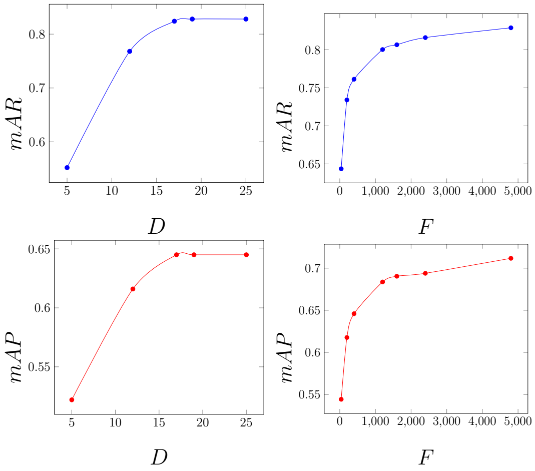

Figure 9: Evaluation results. (*Column 1:*) effect of the tree depth (D) in a Forest on average recall (mAR) and precision (mAP) measures. (*Column 2:*) effect of the number of training frames (F) on mAR and mAP of pixelwise object class segmentation. For the evaluation, 65 real-world test depth maps were used.

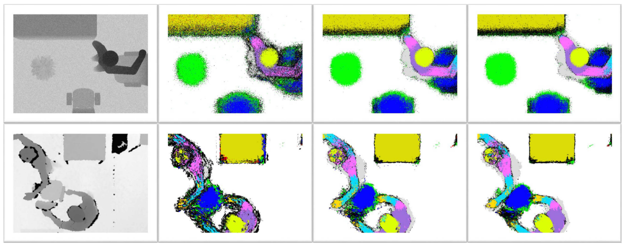

Figure 10: Prediction results based on synthetic and real-world test data for number of training frames. The first column shows the test depth data, and second, third, forth columns show the corresponding prediction results respectively for F={40, 1600, 4800} with probability thresholding of 0.4. Class predictions with a probability less than the thresholds are colored black in the result images.

For the classifier training with the chosen fixed parameter setup resulted in 2.076×10^6 synthetic labeled training samples per tree, with a training time of approximately 43 minutes for a forest and calculating the pixelwise predictions using the trained forest took 34 ms. The testing time is well convenient for the target application and supports real-time processing.

Our proposed approach uses 1600 frames for building a RDF tree where 300 depth values are extracted for each of 10 object-classes in order to compute features specific to a particular object-class for training the tree. This is sufficient for producing almost comparable results to (Shotton et al., 2013). In (Shotton et al., 2013), the authors use 300K frames per tree and 2000 pixels per object-class which takes a high computation cost and a large memory consumption. For the classifier training, our optimal parameter setup resulted in a training and testing time of approximately 43 minutes and 34 ms on a desktop with Intel i7-2600K CPU at 3.40GHZ. In (Shotton et al., 2013), for training 3 trees to depth 20 for 1 million images took time of approximately a day on a 1000 core cluster and 40 seconds of testing time on a 8 core processor. Our work can distinguish subtle changes such as crossed-arms, which is not possible in (Shotton et al., 2013).

5.3 Full Method

The numbers presented in the confusion matrix-based quality measures (Table. 1) are for the RDF classifier and the CRF extension (optimal predictions obtained using α-Expansion based graph cuts extended over RDF predictions), and the pixelwise prediction results illustrated in Fig. 11 are based on the same trained decision forest. For the quantitative evaluation, we use a random number of 65 real-world test depth maps with all object classes.

As a baseline, we implemented the same state-of-the-art (SOA) RDF classifier as used in (Shotton et al., 2013) for pixelwise labeling performance evaluation based on top-view, and compared with our CRF-extension, using a real-world test data. Table. 1 shows the prediction results for the RDF classifier, and the improved prediction results with CRF extension for pixelwise object class segmentation. It can be observed that the CRF extension improves the performance measures by approximately 6.9% in mAR, 19.9% in mAP, and 10.8% in F1-measure over the RDF performance measures. Table. 1 shows a mAP of **0.620**, mAR of **0.816**, and F1-measure of **0.734** are achieved using the RDF classifier, while **0.819** mAP, **0.885** mAR, and **0.842** F1-measure are achieved using the CRF extension. As expected, the results improve using the CRF extension, over RDF classifier.

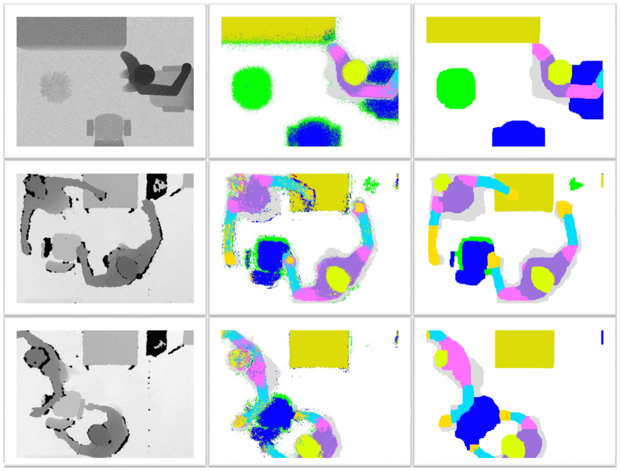

Figure 11: Prediction results based on synthetic and real-world test depth data. The first row is based on synthetic test data, the second and third rows are based on real-world test data. The first column shows the test depth maps, the second and third columns show the predictions obtained from RDF and the CRF extensions.

As a baseline with (Shotton et al., 2013) (Ganapathi et al., 2010) (Dittrich et al., 2014), we compare our performance results using "*top-view*" as a comparison parameter for only human body-parts classification. Fig. 12 shows comparison of the per-joint proposals of the human body-parts classification. Our results for per-joint classification of human body-parts are comparable to (Shotton et al., 2013) and (Ganapathi et al., 2010), and we improve over (Dittrich et al., 2014). Also we obtain comparable results in a faster way than (Shotton et al., 2013). We can see that the mAP of legs is substantially low. We believe that from "*top-view*", legs object class is least discriminative because the industrial grade components fall under the same range of depth measurements as legs (i.e. between foot-waist) object class. In future work, we will be working on improving discriminating legs from other objects in a better way.

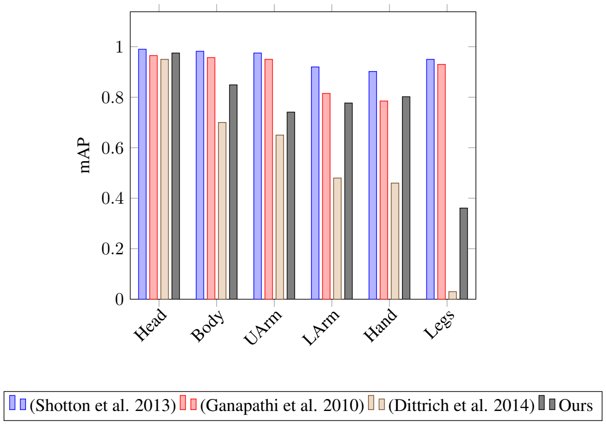

Figure 12: Comparision with (Shotton et al. 2013), (Ganapathi et al. 2010) and (Dittrich et al. 2014). Our approach is sufficient for producing almost comparable results for localizing the joints of the human body-parts.

6 Conclusions and Future Work

We propose a generic classification for pixelwise object class labeling framework. The work is applied to real-time labeling (or segmentation) in RGB-D data from a KINECT sensor mounted on a ceiling placed at the height of 3.5 meters. An optimal and robust parameter setup for pixelwise object class segmentation in real-time and high performance scores are achieved in the evaluation. It is found that increasing the number of training frames (F) monotonically increases the testing prediction. The CRF extension improves the performance measures by approximately 6.9% in mAR, 19.9% in mAP, and 10.8% in F1-measure over the RDF performance measures. In (Shotton et al., 2013), the authors "*fail to distinguish subtle changes in the depth image such as crossed arms*", this is solved by using our training dataset based on "*top-view*". It was demonstrated that the developed approach is relevant, robust and well adapted to the application targeted for pixelwise object class segmentation in industrial domain with humans and industrial-grade components. In future work, concerning human safety, the pose and position of human is very important to be correctly estimated from real-time vision. Based on that, together with a proactive task and path planner a safe human-robot collaboration is feasible as the future goal. Such that, a pure collaboration in common shared area, with common shared tasks can be expected.

Acknowledgements

This work is supported by the BMBF funded project AMIKA and the EU project ROVINA.

References

Fraunhofer IFF (2015). Unit: Human-Robot Interaction.

Table 1: Confusion matrix based mean average recall, precision, and F1-measures

	Avg	Head	Body	UArm	LArm	Hand	Legs	Chair	Plant	Storage	Table
SOA$-RDF_{mAR}$	**0.816**	0.931	0.795	0.718	0.612	0.699	0.972	0.705	0.970	0.930	0.930
SOA$-RDF_{mAP}$	**0.620**	0.971	0.632	0.718	0.709	0.639	0.238	0.941	0.413	0.948	0.948
Ours$-CRFextension_{mAR}$	**0.885**	0.946	0.835	0.849	0.651	0.791	0.987	0.960	0.974	1.0	1.0
Ours$-CRFextension_{mAP}$	**0.819**	0.975	0.849	0.741	0.777	0.802	0.361	0.919	0.846	0.977	0.944
SOA$-RDF_{F1-measure}$	**0.734**	0.950	0.704	0.718	0.656	0.667	0.382	0.806	0.579	0.938	0.938
Ours$-CRF_{F1-measure}$	**0.842**	0.960	0.841	0.791	0.708	0.796	0.528	0.939	0.905	0.988	0.971

Vivek Sharma, Sule Yayilgan, and Luc Van Gool (2015). Scene Modeling using a Density Function Improves Segmentation Performance. *KU Leuven, Technical Report*.

Yuri Boykov and Gareth Funka-Lea (2006). Graph cuts and efficient n-d image segmentation. *IJCV*.

Yuri Boykov, Olga Veksler, and Ramin Zabih (2001). Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI*.

Frank Dittrich, Vivek Sharma, Heinz Woern, and Sule Yayilgan (2014). Pixelwise object class segmentation based on synthetic data using an optimized training strategy. In *ICNSC*.

Marc Freese, Surya P. N. Singh, Fumio Ozaki, and Nobuto Matsuhira (2010). Virtual robot experimentation platform v-rep: A versatile 3d robot simulator. In *SIMPAR*.

Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun (2010). Real time motion capture using a single time-of-flight camera. In *CVPR*.

Josep M. Gonfaus, Xavier Boix Bosch, Joost van de Weijer, Andrew D. Bagdanov, Joan Serrat Gual, and Jordi Gonzalez Sabate (2010). Harmony potentials for joint classification and segmentation. In *CVPR*.

Helmut Grabner, Juergen Gall, and Luc Van Gool (2010). What makes a chair a chair? In *CVPR*.

Xuming He, Richard S. Zemel, and Miguel Carreira- Perpinan (2004). Multiscale conditional random fields for image labeling. In *CVPR*.

Yun Jiang, Marcus Lim, and Ashutosh Saxena (2012). Learning object arrangements in 3d scenes using human context. In *ICML*.

Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees (2006). *IEEE Trans. PAMI*.

Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*.

Jamie Shotton, Ross B. Girshick, Andrew W. Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, and Andrew Blake (2013). Efficient human pose estimation from single depth images. *IEEE Trans. PAMI*.

Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena (2011). Human activity detection from rgbd images. In *AAAI Workshop*.