Human Activity Recognition in the Context of Industrial Human-Robot Interaction

Alina Roitberg†, Alexander Perzylo*, Nikhil Somani*, Manuel Giuliani*, Markus Rickert*, and Alois Knoll†
†Technische Universität München, Bolzmannstr. 3, 85748 Garching, Germany
*fortiss GmbH, Guerickestr. 25, 80805 Munich, Germany

***Abstract*—Human activity recognition is crucial for intuitive cooperation between humans and robots. We present an approach for activity recognition for applications in the context of human-robot interaction in industrial settings. The approach is based on spatial and temporal features derived from skeletal data of human workers performing assembly tasks. These features were used to train a machine learning framework, which classifies discrete time frames with Random Forests and subsequently models temporal dependencies between the resulting states with a Hidden Markov Model. We considered the following three groups of activities: *Movement*, *Gestures*, and *Object handling*. A dataset has been collected which is comprised of 24 recordings of several human workers performing such activities in a human-robot interaction environment, as typically seen at small and medium-sized enterprises. The evaluation shows that the approach achieves a recognition accuracy of up to 88% for some activities and an average accuracy of 73%.**

I. Introduction and Motivation

Human-robot interaction (HRI) is often researched in the context of domestic applications or social studies. The application of HRI in industrial domains differs in several important aspects. One of these aspects is the types of activities which are recognized, which are very different for service / personal robotics and industrial robotics. Also, the working environment and the contained objects are very different in such scenarios. In this paper, we focus on activity recognition suitable for small and medium-sized enterprises (SME). In contrast to large-scale industries, production processes in SMEs are less structured and change more frequently, which requires higher flexibility and reconfigurability from the used HRI system. These systems should enable their operators to easily set up new tasks and they should be able to some extent to adapt to unforeseen situations. Human activity recognition can be used by HRI systems to assess the current state of interaction with the human and to predict the following events in the work flow.

In this paper we present a machine-learning framework for human activity recognition in an industrial HRI workcell and a systematic evaluation of different machine learning techniques, which have been tested on this framework. We recorded and manually annotated RGB-D videos and skeletal data of multiple humans performing typical activities related to manufacturing in SME-like environments. Figure 1 depicts the used testbed. Human activities were grouped into three classes: *Movement* activities (e.g. *Entering* and *Leaving* the scene), *Gestures* and communication signals (e.g. *Pointing* to and *Presenting* objects), and *Object handling* (e.g. *Grasping* and *Moving* an object). For collecting data on the human's actions, we use two synchronised Microsoft Kinect RGB-D sensors, which are installed on the HRI workcell in a configuration in which they cover the whole work space. The gained data is used to train a machine learning framework for human activity recognition. We compared different machine learning algorithms using leave-one-out cross-validation and found that a combination of Random Forests and Hidden Markov Models (HMM) performs best.

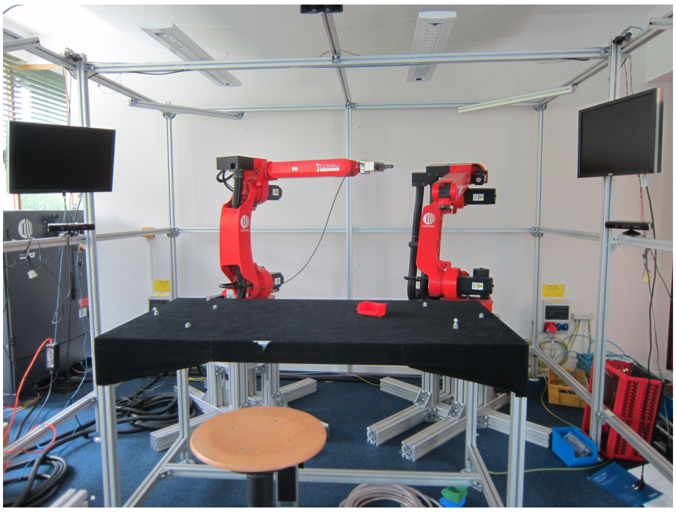


Fig. 1: The SME-like robotic workcell for human-robot cooperation consists of two industrial robot arms, a table and a cage equipped with 2 Kinect sensors. The monitored work space is shared with a human to work on assembly tasks.

The main contributions of this paper are: (1) the development of a human activity recognition framework in the context of industrial HRI scenarios, which uses a two-step approach for the recognition of spatio-temporal activities, (2) the design of feature vectors which are suitable for recognition of such activities, and (3) a comparison of different machine learning algorithms for feature importance estimation and classification.

II. Related work

A good portion of recent research in computer vision has focused on detection of human activity, mostly with data obtained from 2D videos [17], [22], [13]. The recent popularity of affordable RGB-D sensors such as Microsoft Kinect or Asus Xtion made it possible to collect high quality 3D data and lead to development of skeleton tracking software such as Nite and Kinect SDK. Following the work of Shotton et al. [19], which introduced a robust real-time method for skeleton capturing with random forests, the usage of 3D skeleton data for activity recognition gained popularity in the past few years [14], [16]. Machine learning methods that were applied to activity recognition include both discriminative classifiers such as support vector machines, k-nearest neighbours, and random forests [1], [2], [4], [25] as well as generative approaches such as Hidden Markov Models (HMMs). In our work, we use HMMs for modelling and recognising activity sequences. Yamato et al. [24] were the first who proposed to use discrete HMMs for human activity recognition. In their work, they applied a clustering algorithm on the vectors of image features and then used centroid of the corresponding cluster as input symbol for the HMM. Similar to Yamato et al's work, we combine a classification algorithm and train a Hidden Markov model with the corresponding labels as input. Instead of unsupervised clustering, we apply a supervised classification algorithm for quantisation and subsequently use an HMM to represent temporal relationships between different activities.

Similar to our work, researchers in activity recognition recorded datasets for training and testing their recognition methods. The Cornell Activity Dataset (CAD) [20], [21] is one example for a dataset that was used by several researchers. The CAD-60 dataset includes 12 activities from five different domestic environments, for example cooking or brushing teeth. Sung et al. [20], [21] proposed to decompose complex activities into sub-activities and to use a maximum entropy Markov model for recognition. With this method the authors reached the precision/recall values of 67.9%/55.5% with a person not present in the training set. Koppula et al. [9], [8] reported significantly improved recognition results (precision/recall 80.8%/71.4%) when using structured support vector machines on the same data set. The MSRAction3D dataset [11] contains recordings of 20 activities, such as hammering, drawing, and hand clapping. The MSRDaily3D dataset [23] covers daily human activities, for example drinking, writing, and using a laptop. The MSRPair3D dataset [15] contains pairs of actions, including picking up and putting down a box. In our work, we are using a data set that we recorded ourselves, because there are no activity datasets available for our target scenario, the cooperation of humans and robots in an industrial context.

The potential of using human gestures in context of industrial HRI has been discussed by Gleeson et al. [5]. The authors have designed an experiment to identify the common gestures that would be naturally used by human workers during collaboration on a defined assembly task. The experiment has shown that body gestures is a highly efficient way of interaction, though the interpretation is highly context sensitive. Based on the observations the authors have created a lexicon of most useful hand gestures, which also included multiple gestures used in our experiment (e.g. pointing, presenting). These findings demonstrate the importance of realizing HRI systems based on body gestures, which is also the focus of our work.

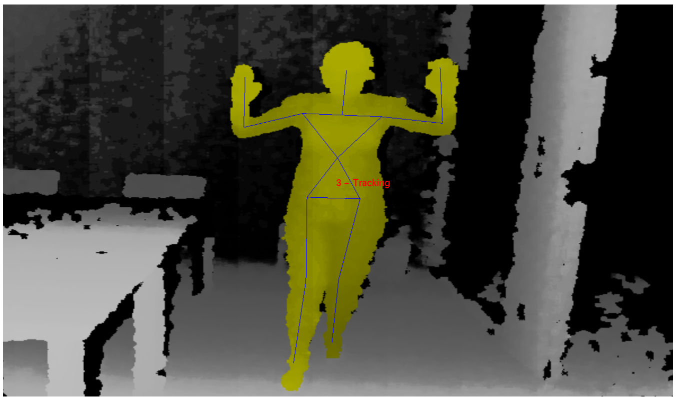


Fig. 2: Fifteen skeleton joints provided by the NITE skeleton tracker.

Despite of the findings of Gleeson et al., there is a lack of work focused on activity recognition in industrial environments. The work most similar to ours is that of Lenz et al. [10]. The authors automatically recognised the activities of humans, who were sitting in front of a work table and collaborated with an industrial robot. They tracked the positions of the humans' hands using 3D occupancy grids and used a composite HMM for recognition of hand gestures and work-flow analysis.

III. Human Activity Recognition

In this section, we describe our approach for human activity recognition. Section III-A describes the HRI system that we used for recording the activity data. Section III-B gives details about the used RBG-D sensors, their calibration and synchronisation. The activities and scenarios that we used for the recordings are described in sections III-C and III-D. Finally we give technical details about the used features and machine learning model in sections III-E and III-F respectively.

A. Human-robot interaction system

For the activity recordings, we recreated a typical environment of a human-robot cooperation in the context of an SME. Figure 1 shows a picture of the environment, which consists of a working table with a human on one side and two industrial robots assisting the worker from the other side of the table. The human can either stand or sit in front of the table. As manufacturing work flows usually include working with tools (sawing, drilling etc.), there were several objects, including work tools and coloured boxes, placed on the work table.

The setup also contains a metal frame that surrounds the work table and the robots and can be used to place sensors and displays in the environment. We placed two Microsoft Kinect RGB-D sensors on the metal frame above the left and right corners of the table, facing the typical worker position at an angle of approximately 45 degrees. The best calibration pose for skeleton tracking is one facing towards the sensor at a distance of approx. 2.5m. Hence, placing the sensors in this configuration allows more flexibility during the entering part, as usually one of the sensors would track the skeleton, depending on the side from which the human enters.

B. *Sensor calibration and synchronisation*

As mentioned above, we used two Microsoft Kinect RGB-D sensors. For image segmentation and person tracking, we used the Nite skeleton tracking algorithm[1] that provides the poses of fifteen joints for each tracked human. Figure 2 shows a picture of the skeleton joints. Kinects capture RGB data with a 1280x960 pixel resolution as well as the corresponding depth map depicting. We recorded depth and colour frames at 30 Hz frequency, which is also the frequency of the detected skeletal frames.

Combining multiple Kinect sensors: Conditions for ideal skeleton tracking with RGB-D sensors include full body visibility, as well as facing the sensor at a distance of approximately 2.5 meters. When using such sensors in complex environments, joint occlusions often become a problem. This is especially true in an industrial setup, where the person focuses on the work table and is rarely looking directly at the sensor, while the lower body is completely covered by the table. Obtaining high quality skeleton data in such settings is very challenging. To increase the robustness and flexibility of the system, two Microsoft Kinect sensors were combined to increase the field of view. The sensors are facing the position of the worker at approximately 45 degrees from the left and the right side. The sensors were calibrated extrinsically using images of a planar checker board pattern as a reference grid with Camera Calibration Toolbox for Matlab[2]. The coordinates of the skeletons of each Kinect were transformed with corresponding translation and rotation to the common coordinate system, with the point of origin placed approximately at the ideal workers position.

As both sensors together cover almost all of the possible movement-space, a person entering the scene in the calibration pose is quickly tracked by one of the sensors (depending on the side from which the person enters). If the human is tracked by only one sensor (depending on the entrance point), the skeleton is transformed to the common coordinate system and directly used. In the case where the user is tracked by both sensors, we select one of them and use its tracking results as long as it provides good quality data (based on the confidence values). The tracking sensor is switched only if the current device loses the skeleton or has not provided any data with satisfying confidence (i.e. at least one joint of the upper body part with 1.0 confidence value) during the last ten frames. This method assures continuous motion of the joints which is interrupted only if the currently used sensor is not tracking the skeleton correctly.

The common issue of the interfering infra-red patterns when combining multiple depths sensors [12] was not significant for our placement of the sensors and no significant decline in quality of skeleton tracking was observed.

C. *Recorded Activities*

In order to train and test our human activity recognition approaches, we recorded typical activities related to humans working in SME environment. We divided these activities into three groups: *Movement*, *Gestures* and *Object Handling*. The activities in each group are independent from activities in other groups and can occur simultaneously. More formally, this can be expressed as: at each time instance of interaction with the HRI system, the activity state of the person can be determined as a triplet (M, G, O), with M, G and O being the states belonging to the *Movement*, *Gestures* and *Object Handling* activity groups respectively. Table I gives an overview of all activities and their relation to each of the recording scenarios (Section III-D). The ground truth activity label for each group was added to every skeleton frame.

	Scenario number				
Activity	**1**	**2**	**3.1**	**3.2**	**3.3**
Movement					
Entering	✔	✔	✔	✔	✔
Leaving	✔	✔	✔	✔	✔
Stepping back			✔	✔	✔
Sitting			✔	✔	
Standing	✔	✔	✔	✔	✔
Leaning Forward			✔	✔	✔
Gestures					
Pointing to an object		✔			
Presenting an object	✔				
Seeking attention (waving)			✔	✔	✔
Interrupt / Stop action			✔	✔	✔
None	✔	✔	✔	✔	✔
Object handling					
Reaching	✔	✔			
Grasping / Picking	✔	✔			
Moving	✔	✔			
None	✔	✔			

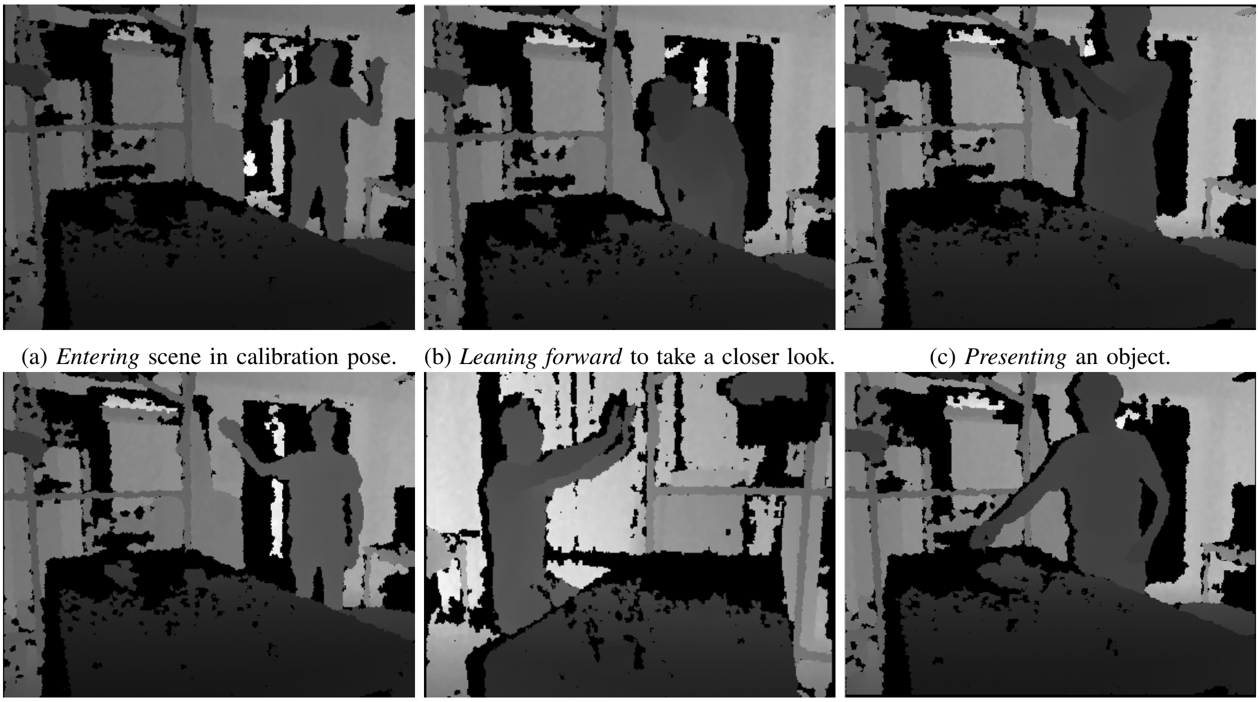
TABLE I: Overview of activities and scenarios used for data recordings.

1) Movement Activities: These activities include *Entering* the scene, *Leaving* the scene, *Stepping back* from the table, *Sitting* at the table, *Standing* at the table, and *Leaning forward* to take a closer look at an object. We defined *Entering* the scene as entering the environment in PSI-calibration pose and moving towards the table. The transition from *Entering* to *Standing* happens when a person stops moving and is no longer in calibration pose. *Leaving* the scene usually began with turning around. In some cases, however, the person took several steps back before the turn, which made the distinction between the states *Leaving* and *Stepping back* difficult. Figures 3a and 3b show example depth images for activities *Entering* and *Leaning forward*, respectively.

2) Gesture Activities: This group of activities covers gestures and represents the signals that the worker would send to the robot in explicit communication. Gestures included *Pointing* to or *Presenting* an object to a robot, *Seeking attention* (waving the hand), *Interrupt/stop* the robot's action (both hands in front of the torso in a "stop" position) and *None* (default activity). Figures 3c, 3d, and 3e show example depth images for activities *Presenting*, *Seeking attention*, and *Interrupt/stop*, respectively.

[1] http://www.openni.org

[2] http://www.vision.caltech.edu/bouguetj/calib_doc/



(a) *Entering* scene in calibration pose. (b) *Leaning forward* to take a closer look. (c) *Presenting* an object.

(d) *Seeking attention* (waving). (e) *Interrupt/stop* signal. (f) *Reaching* for an object.

Fig. 3: Depth images of selected activities.

3) Object Handling Activities: The third group includes all activities related to object handling. This includes *Reaching*, *Grasping*, *Moving*, and *Placing* an object. Again, we added the *None* state, which represents not handling an object. The activities in this group are causally dependant on each other, *Reaching* is the predecessor of *Grasping*, followed by *Moving* and *Placing*. Figure 3f shows an example depth image for the *Reaching* activity.

D. Scenarios

We defined different scenarios to create appropriate sequences of activities. Each scenario shows one person entering the scene in the calibration pose, heading towards the work table, and stopping in front of it. The person then performs a number of actions, either sitting or standing in front of the work table and leaves the scene in the end. Table I shows an overview of the activities which can be seen in each scenario.

In the first scenario, the person reaches for an object, grasps it, presents it to the robot and puts it back down.

In the second scenario, the person reaches for an object on the work table, picks it, and places it at a different location on the table. The person then points to a box on the work table. This action is an example for a signal to the robot for performing further actions on an object.

The third scenario contains the longest chain of activities and has several variations. The activities include seeking for attention of the robot by waving the hand, interrupting the robot's action, leaning forward to take a closer look at an object on the work table, and taking a step back. The activities were performed in randomised order, either sitting on a chair or standing in front of the work table. In one of the variations the person performs the seeking attention activity after taking a step back, being slightly distant from the table.

E. Features

The input of our activity recognition framework are RGB-D images obtained from two Kinect sensors. Skeletons are detected in each of these images using the NITE skeleton tracking framework, which models a skeleton as a set of positions and orientations of fifteen joints with respect to the sensor as well as a set of confidence values, which can be 1 (tracking is working), 0.5 (joint configuration was inferred from the skeleton heuristics) or 0 (tracking fails).

Raw skeletal information requires further processing. In the SME-like work environment that we created for this paper, the legs of the person in front of the sensors were rarely visible, because they were occluded by the working table. Therefore, the position information about the person's legs that NITE computes is wrong most of the time. For this reason, we decided to completely dismiss leg information and put a stronger focus on the position of the hands. Additionally, the 3D coordinates of all joints are given w.r.t. the sensor position, which disturbs the inference of true body posture. We used torso position to characterise the position of the person relative to the point of origin, and the coordinates of all other joints are calculated w.r.t. this frame. The extracted features are similar to the ones proposed by Sung et al. in the Cornell Activity Recognition project [20], [21]. The next paragraphs give a more detailed description of all features.

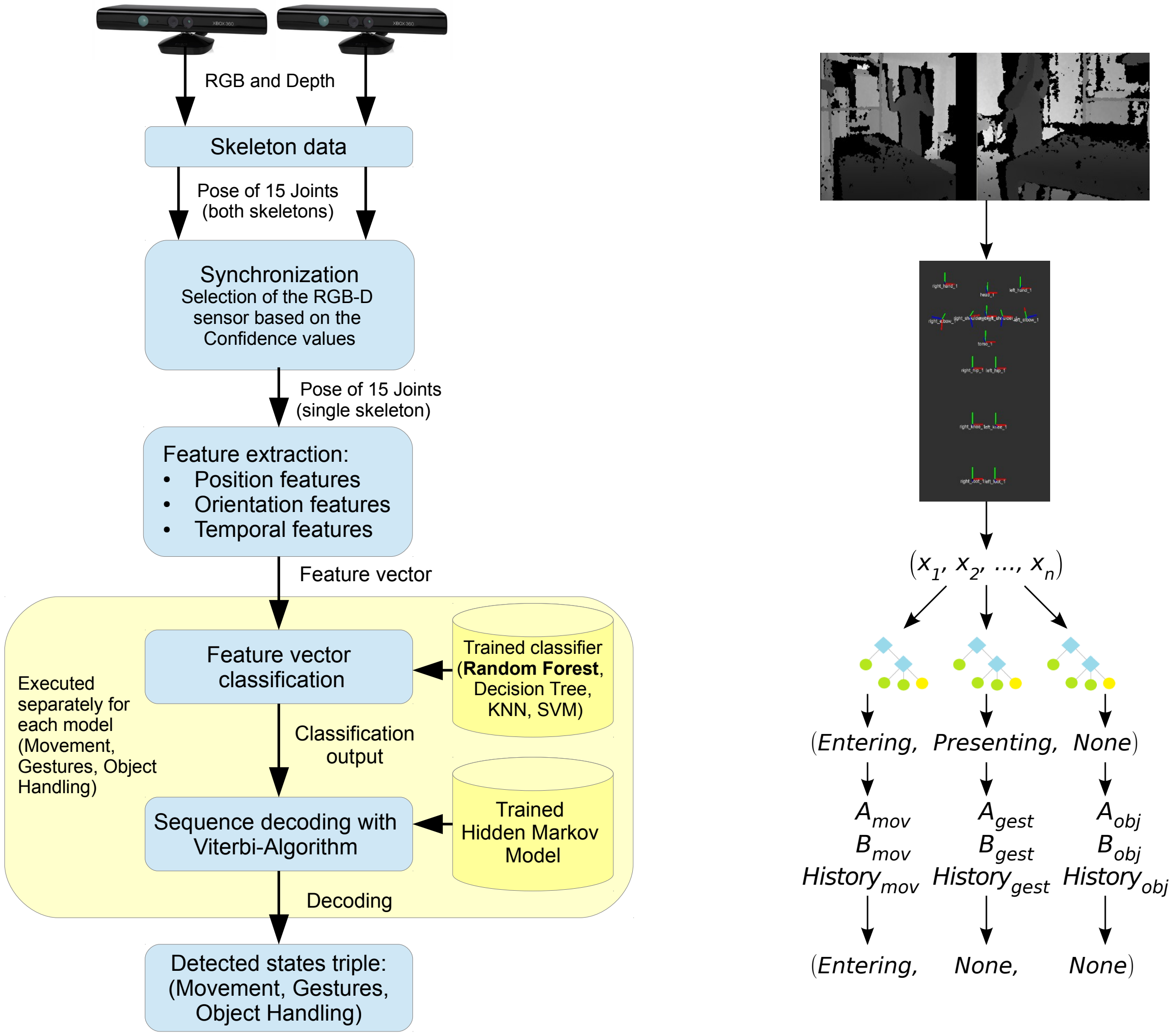

RGB and Depth
Skeleton data
Pose of 15 Joints (both skeletons)
Synchronization
Selection of the RGB-D sensor based on the Confidence values
Pose of 15 Joints (single skeleton)
Feature extraction:
• Position features
• Orientation features
• Temporal features
Feature vector
Executed separately for each model (Movement, Gestures, Object Handling)
Feature vector classification
Trained classifier (**Random Forest**, Decision Tree, KNN, SVM)
Classification output
Sequence decoding with Viterbi-Algorithm
Trained Hidden Markov Model
Decoding
Detected states triple: (Movement, Gestures, Object Handling)
$(x_1, x_2, ..., x_n)$
(Entering, Presenting, None)
A_{mov} B_{mov} $History_{mov}$
A_{gest} B_{gest} $History_{gest}$
A_{obj} B_{obj} $History_{obj}$
(Entering, None, None)

Fig. 4: Proposed machine learning framework (left) and an example showing the corresponding recognition process for the *Entering* activity (right)

Position features: we define the body position of the person as the coordinates of the torso joint w.r.t. the point of origin. As a lot of attention is drawn to the pose of hands, we calculate the pose of both hands and elbows w.r.t the torso as well as the hand pose w.r.t. the elbow.

Orientation features: the orientation of upper body part provides information about the direction of the person's field of view. Consequently, we calculate the angle between hip line and shoulder line to estimate if the person has turned or not. We also estimate body orientation in relation to the camera, which is important for movement activities such as *Entering* and *Leaving*. The angle of the vertical body line, connecting head and torso, helps distinguish if the person is leaning forward or standing straight. The angle of both elbows as well as the angle of the arm relative to the body is used to determine the position of the hands.

Temporal features: due to the dynamic nature of human activities, analysing the velocity of the joints is crucial. We use the skeletal configurations recorded with a delay of 0.5s, 1s, 1.5s, 2s, and 3s before any given feature set to calculate the velocity of the body position, with direction and magnitude as separate features, and the hand and elbow velocity as well as the angular velocities of the orientation features.

F. Machine Learning Framework

Recognizing human activities based on body posture is a complex task and has to be considered from various aspects. On the one hand, looking at one video frame already gives the information about body posture from which we can derive the most likely activity by considering what the feature vector looks like. Such classification methods are widely used in activity recognition, with Support Vector Machines and K-Nearest-Neighbours being the most prominent approaches [6]. Another important aspect of activity recognition are causal and temporal relationships between different activities, which have to be modelled as part of the recognition approach. The probability of a video frame depicting a certain action also depends on the state in which the human has been in the previous frame, making it crucial to analyse the evolution of activities over time.

An excellent way to model the sequential processes that we are facing in human activity recognition are Hidden Markov Models (HMM), which are widely used in speech and video recognition. As stated by Rabiner[18] a discrete HMM is given by an alphabet of possible hidden states and a set of possible observations. At each time point, current state makes a transition to a different state with a certain probability and produces a visible emission. Assuming there are n hidden states and m possible emissions, an HMM can be specified as a triplet (π, A, B) where A is a $n \times n$ transition matrix containing transition probabilities between the states, B is the $n \times m$ emission matrix with $B(i, j)$ showing the probability of a hidden state i producing a visible emission j and π being the initial probability vector of hidden states.

HMMs can be used for a variety of problems, and in our application the *decoding* and the *learning* problems are significant.

- Decoding: It is the estimation of most likely sequence of hidden states that took place with a given model (π, A, B) and an observed sequence of emissions X. This problem can be solved with the Viterbi Algorithm with run-time complexity of $O(n^2T)$, where n is the number of hidden states and T the length of the sequence.
- Learning: It is the estimation of model parameters (π, A, B). Here one should distinguish between supervised learning, where the corresponding sequence of states is known, and unsupervised learning, when only the emissions are given. In case of supervised learning the statistics of known samples can be used to calculate the transition and emission probabilities. In case of unsupervised learning, Baum-Welch expectation-maximization algorithm can be used to optimize the model parameters after an initial guess of transition and emission probabilities.

For our activity recognition approach, we propose a two-stage machine learning framework. Figure 4 shows a graphical representation of the processing steps in this approach. In the first stage, a supervised machine learning algorithm was applied on a subset of the input feature vectors (described in III-E) and a corresponding annotation to classify the data. Thus, the input training data was a set of the data samples, in which every sample consisted of the feature vector calculated from a skeleton frame and the manually added annotation of this frame. To reduce bias towards over-represented classes, which in our case is mostly the *None* activity, the training data was randomly sampled such that there were equal number of samples in each class. Different supervised learning algorithms (KNN, SVM with Kernel, Decision Trees, Random Forests) were considered for classification (see Sec. IV-D), and **random forests** were chosen for the final model, since they had the best performance. Figure 5 shows the results of feature importance estimation using a forest of 30 randomised trees.

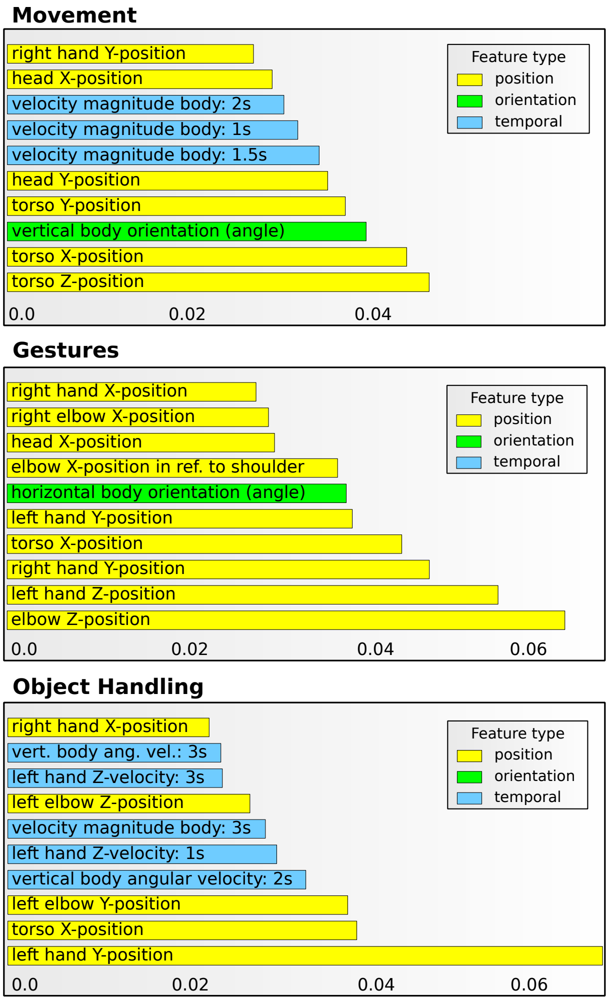
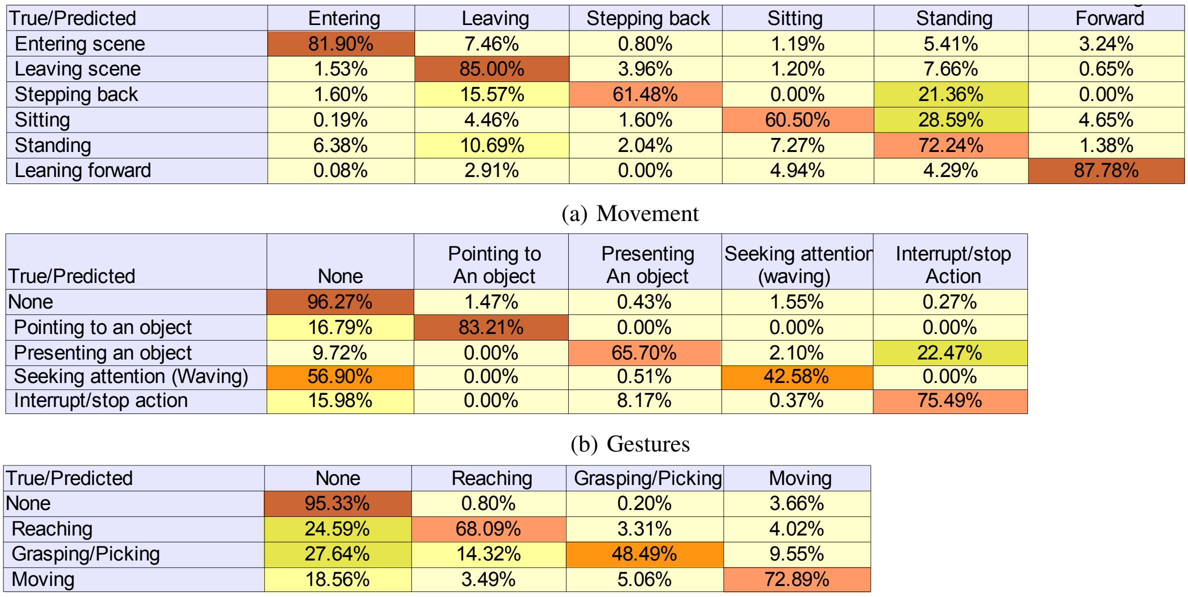


Fig. 5: Feature importances estimated with forest of 30 randomised trees (top ten features).

The second stage of training aims at modelling temporal dependencies between different states with a **HMM**. This grants control over causal relationships and allows restrictions to the flow of actions. For example, action *Moving* an object cannot take place without previous actions *Reaching* and *Grasping*. In this phase of training each recording is used as an input sequence for training of an HMM. For each video frame, the feature vector was calculated, classified with the algorithm from the first phase, and used as an emission input to train the HMM. The training of HMM estimates the transition matrix A, where $A(i, j)$ is the probability of state i changing into state j in the next frame, and emission matrix B, with $B(i, j)$ depicting the probability, that state i produces a feature vector that will be classified as j in the first stage.

The models for activities of the three activity groups *Movement*, *Gestures*, and *Object handling* were trained separately and are independent. Consequently, the classification framework produces a triplet (M, G, O), where M is a state representing a *Movement* activity, G represents a *Gesture*



True/Predicted	Entering	Leaving	Stepping back	Sitting	Standing	Forward
Entering scene	81.90%	7.46%	0.80%	1.19%	5.41%	3.24%
Leaving scene	1.53%	85.00%	3.96%	1.20%	7.66%	0.65%
Stepping back	1.60%	15.57%	61.48%	0.00%	21.36%	0.00%
Sitting	0.19%	4.46%	1.60%	60.50%	28.59%	4.65%
Standing	6.38%	10.69%	2.04%	7.27%	72.24%	1.38%
Leaning forward	0.08%	2.91%	0.00%	4.94%	4.29%	87.78%

(a) Movement

True/Predicted	None	Pointing to An object	Presenting An object	Seeking attentior (waving)	Interrupt/stop Action
None	96.27%	1.47%	0.43%	1.55%	0.27%
Pointing to an object	16.79%	83.21%	0.00%	0.00%	0.00%
Presenting an object	9.72%	0.00%	65.70%	2.10%	22.47%
Seeking attention (Waving)	56.90%	0.00%	0.51%	42.58%	0.00%
Interrupt/stop action	15.98%	0.00%	8.17%	0.37%	75.49%

(b) Gestures

True/Predicted	None	Reaching	Grasping/Picking	Moving
None	95.33%	0.80%	0.20%	3.66%
Reaching	24.59%	68.09%	3.31%	4.02%
Grasping/Picking	27.64%	14.32%	48.49%	9.55%
Moving	18.56%	3.49%	5.06%	72.89%

(c) Object handling

Fig. 6: Confusion matrices for each activity group validaton.

activity, and O represents an *Object handling* activity. After the model is trained, activity recognition of a sequence of skeleton-frames can be performed by classifying the frames with the learned model (in our case with random forest approach), using the classified data as input to the estimated HMM, and decoding it with the Viterbi algorithm to produce the final labels (M, G, O).

IV. Evaluation

A video showcasing results from the activity recognition framework can be found online[3].

A. Dataset

The framework was evaluated with 24 recordings (7 times scenario 1, 7 times scenario 2 and 10 times scenario 3 in different variations), with three different people performing the actions. Most of the target activities occur in multiple scenarios (e.g. entering, moving an object) while others take place in only one (e.g., sitting). The sessions were recorded as ROS-bag files, containing frames of skeleton information of both Kinect sensors, annotation (three ground-truth states in each frame) as well as colour and depth images. Please note that we did not use the colour and depth images in this work, but will use it in future work. Ground-truth annotation was manually added to the skeleton frames. The whole test set contains 23710 skeleton frames after synchronisation of both sensors, with approximately 988 frames per recording session on average.

B. Feature Importances

We used random forests to estimate the importances of the features for the classification task. In a decision tree, the depth of a feature can be used to estimate its relative importance, i.e. the expected fraction of samples they contribute to. In a random forest this value can be estimated by calculating the average fraction of wrongly classified samples after deletion of the corresponding feature from the model [3]. Figure 5 shows the results of the feature importance estimation.

As expected, the significant features of the *Movement* activity group are concentrated around the torso position with respect to the sensor and body/torso velocity. The vertical angle of the body (angle between the backbone line and Y-axis) is also significant, which is expected for activities like *Leaning forward*. It is also not surprising, that the most important features of the *Gestures* and *Object handling* activity groups are focused around hands.

An interesting observable difference is the fact that the ten best features of the *Gestures* group belong to the position and orientation group, while velocity features are crucial for *Object handling*. This observation is especially important because the attempt to detect object handling actions without any object or finger tracking was quite experimental and it was questionable if it would be possible to differentiate *Reaching*, *Grasping*, and *Moving* an object from gestures like *Pointing*. The *Reaching–Grasping–Placing* sequence contains continuous and relatively fast movements of the hands, whereas most of the gestures (like *Pointing* or *Seeking for attention*) contained one fast motion at the beginning and the end while being in more or less constant state in between.

It should also be mentioned, that the problematic frames in the *Gestures* activity group tend to occur at the beginning

[3] http://youtu.be/rNDR5EsSNfI

and at the end of the gesture, while the frames in the middle of action are usually recognized correctly. The fact that the velocity features are not characteristic in the trained model is a possible explanation for this phenomenon.

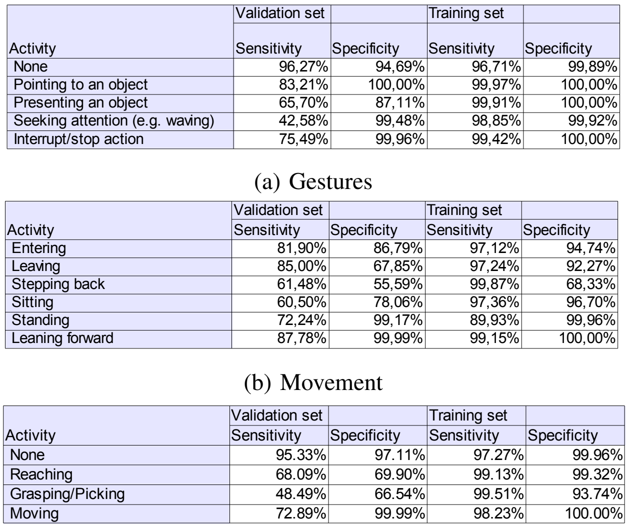
C. Activity Recognition Validation

We evaluated the performance of our activity recognition approach with leave-one-out cross validation (LOOCV), using each of the recordings as evaluation set once. The results depict average values over 24 rounds of cross validation. Figure 6 shows the validation results. We separately evaluated the frame-wise accuracy for each activity and present the results sorted by activity groups.

The *Movement* activity group detection performed the best of the three groups, with most of the labels predicted correctly in each class. Entering and *Leaving* the scene are usually correctly detected (81.9% and 85.0%). The activities *Standing* and *Sitting* are not easy to distinguish, which is justified by the fact that the legs of the recorded persons are not visible in our environment and the height differences between sitting and standing positions are minor for some people involved in the experiment. Nevertheless *Sitting* was correctly classified in 60.5% of the frames while being wrongly labelled as *Standing* in 28.59%. The best detected activity was *Leaning forward* with 87.78% of samples recognized correctly. This was expected, as this activity is very sensitive to the vertical body angle, which is the third most important feature in the *Movement* activity group.

Recognition of the activity *Stepping back* was also problematic, with 61.48% accuracy and high confusion with *Standing* and *Leaving* (21.36% and 15.57% respectively). The activity is problematic, since the frames were annotated as *Stepping back* only if this was an explicit command of the corresponding scenario. Small steps back which might occur while performing tasks were classified as *Standing*. When executing the command *Leaving*, the person turns around immediately in most of the recordings, while some recordings exist where several steps back are taken before the turn. The whole sequence was also annotated as *Leaving*, adding to the confusion.

Results in the *Gestures* activity group were mixed. While the states *None*, *Pointing*, *Interrupt/stop* and *Presenting an Object* performed well (accuracies 96.27%, 83.21%, 75.49% and 65.7%, respectively), *Seeking attention (waving)* was usually misinterpreted (accuracy 42.58%). Nearly all activities in this group, with an exception of the *Presenting an object* activity, have shown a bias towards the *None* activity, while almost never being confused with each other. The change of body posture during hand gestures often contains three phases: a fast movement at the beginning (moving the hands to the corresponding position), a constant phase (e.g., pointing to an object for several seconds) and another fast position change in the end as the "return" to normal position. These phases are also called *preparation*, *nucleus*, and *retraction* in gesture literature[7]. During the annotation, the whole movement sequence of a gesture was classified as the corresponding gesture, which might be confusing, as start and end phases of such activities are very similar to each other and as well as to object handling activities. When analysing video sequences, the "constant" part of gestures was almost always recognised correctly, while problems occurred when lifting up and putting down the hands. This observation is also consistent with the previous observation of feature importances, in which most significant features of the *Gesture* activity group are static features (not velocity features).



	Validation set		Training set	
Activity	Sensitivity	Specificity	Sensitivity	Specificity
None	96,27%	94,69%	96,71%	99,89%
Pointing to an object	83,21%	100,00%	99,97%	100,00%
Presenting an object	65,70%	87,11%	99,91%	100,00%
Seeking attention (e.g. waving)	42,58%	99,48%	98,85%	99,92%
Interrupt/stop action	75,49%	99,96%	99,42%	100,00%

(a) Gestures

	Validation set		Training set	
Activity	Sensitivity	Specificity	Sensitivity	Specificity
Entering	81,90%	86,79%	97,12%	94,74%
Leaving	85,00%	67,85%	97,24%	92,27%
Stepping back	61,48%	55,59%	99,87%	68,33%
Sitting	60,50%	78,06%	97,36%	96,70%
Standing	72,24%	99,17%	89,93%	99,96%
Leaning forward	87,78%	99,99%	99,15%	100,00%

(b) Movement

	Validation set		Training set	
Activity	Sensitivity	Specificity	Sensitivity	Specificity
None	95.33%	97.11%	97.27%	99.96%
Reaching	68.09%	69.90%	99.13%	99.32%
Grasping/Picking	48.49%	66.54%	99.51%	93.74%
Moving	72.89%	99.99%	98.23%	100.00%

(c) Object handling

Fig. 7: Validation results (sensitivity and specificity) for each group of activities in training and validation sets.

As shown in Figure 7, the specificity of the *Gesture* activities is very high compared to the *None* activity, which is the lowest one (94%). However, these values should be analysed with care, as some activities (especially the *None* activity) are strongly over-represented. Hence, specificity of 94%, which might seem high at first sight, is quite low due to the fact that most of the time the person was doing nothing. The confusion matrix is a better metric in this case, as it handles confusions of each pair of classes separately.

The recognition results for the *Object handling* activity group were surprisingly high, even with the absence of additional object or finger tracking data. However, the results of the feature importances analysis give us insight about the ways such recognition might work with skeleton data only (position and angular velocities of hands as well as angular velocity of vertical slope of the body were of great importance). The fact that all three activities in this group were a single sequence (*Reaching–Grasping–Moving*) with only one possible order of activity, made the detection with an HMM easier.

Although *Object handling* had lower accuracy compared to *Movement* or *Gesture* detection, most of the samples were labelled correctly, with *None* being correctly recognized in 95.33% of the frames, *Reaching* and *Moving* and *Placing* an object recognized in 68.09% and 72.89% of the samples and

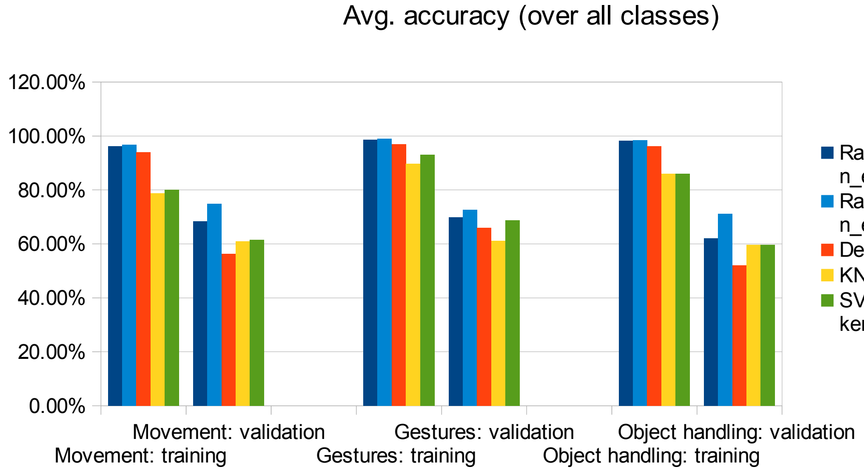


Fig. 8: Comparison of different classification methods used in the first stage of the proposed framework. Note: in each case classification with Hidden Markov Models was subsequently applied (second stage).

Grasping being the activity with the lowest results of 48.49%. A possible explanation for this is the fact, that *Grasping* was a much shorter action then other activities and common confusion on the beginning and the end of the action results in major decline in accuracy. Grasping was also often confused with the similar activities *Reaching* and *Moving/Placing* (14.32% and 9.55% of the frames). All activities are frequently confused with the *None* state. However, contrary to the recognition of gestures, the specificity of the non-default states is not as high as the states are often confused with each other.

Better results in the recognition of the *Movement* activities, followed by *Gestures* and *Object handling* is not surprising. Rough estimation of the skeleton position provides more significant information about large body movements such as *Entering* or *Leaning forward* then about precise movements of the hands and interactions with objects. Although our framework already puts more focus on the arms, more data about the area around the hands such as finger or object tracking might significantly improve the recognition results. Besides this, separate feature selection and reduction for every model would be beneficial as different types of features are relevant for different groups of activities, as discussed in IV-D. As a separate random forest classifier is trained for every model, automatic weighting of the features based on their importance is implicitly included. Nevertheless, the same set of features is currently used for training of the random forests and removal of not informative dimensions for each group separately might lead to better results.

D. *Comparison of different classification methods*

Different classification methods were considered for the first phase of the learning framework, with random forests showing the best results in leave-one-out cross validation, followed by support vector machines with kernel, which is a widely used approach for human activity recognition. The following results compare the performance of the proposed two-staged framework (described in 4) with different machine learning methods used in the first stage, while using the same Hidden Markov Model classification approach for the second stage.

Different metrics can be used used to estimate the overall performance of the model. The most commonly used measure is the overall accuracy, as the fraction of correctly classified frames and total number of the frames. However, this measure is strongly biased towards over-represented classes, which is the *None* state in our case. This is highly undesirable for us, as we are more interested in the detection of “active” actions.

For this reason we also estimated the average accuracy over all classes, for which the accuracy is measured separately for each class and then divided by the number of classes, so that all states contribute equally to the results. In other words, average accuracy over all classes depicts the average value of the confusion matrix diagonal. Figure 8 shows the results of the average accuracies for all activities over all validation rounds.

We compared the performance of random forests with different number of randomised trees and decided to use 30 estimators (Movement: 74.82%, Gestures:72.65%, Object handling: 71.2%), since using 10 estimators showed a decline in performance. Further increase in the number of trees did not result in any significant improvement producing both slightly better and worse results. Support vector machines with radial basis function kernel were the second best approach, showing slight decline in *Gesture* recognition and a stronger decline in *Movement* and *Object handling* recognition (*Movement*: 61.43%, *Gestures*: 68.74%, *Object handling*: 59.62%). The K-nearest-neighbours algorithm with $K = 20$ as well as a single decision tree have shown weaker performance. Nevertheless the average accuracy over all classes is still higher then 50% for each activity group.

V. Conclusion and outlook

In this paper, we presented a two-step approach for activity recognition based on random forests and Hidden Markov Models for use in industrial human-robot interaction scenarios. The recognition is based on skeleton features. The 3D coordinates of skeletal joints were obtained from RGB-D data of two synchronised Microsoft Kinect sensors using NITE skeleton tracking algorithm. We defined different scenario-related activities and grouped them into the three activity groups (*Movement*, *Gestures*, and *Object handling*). As training data, we used recordings of humans performing activities in three varied scenarios, in different order and under different conditions. We extracted position, orientation, and temporal features from the raw skeleton data (positions of fifteen joints), dismissing leg information and putting stronger focus on the hands. We applied random forests on the calculated feature vectors to choose the most important features. Finally, we trained a Hidden Markov Model with the classified skeleton frames. The performance of this system was evaluated based on 24 recordings of different scenarios with leave-one-out cross validation. The results showed average frame-wise accuracy (averaged over all possible classes) of 74.82% for *Movement*, 72.65% for *Gestures* and 71.2% for *Object handling*.

Most of the classification failures of the *Movements* and *Gestures* activity groups occurred either at the beginning or at the end of the corresponding activity, which can be explained by overlapping movements (e.g., *Stepping back* while *Leaving* the scene). Multi-level classification of activities could be used to solve this problem. For example, detection of low-level activities such as stepping back, moving forward or sideways could build the basis for recognising more complex activities like entering or leaving the scene. Also, we will take advantage of more features extracted from RGB-D data. Additional information around the hands and object and finger tracking would be a straightforward improvement for object handling activities. Gaze tracking might be very beneficial for human-robot communication, as it is a direct indicator of what a human's attention is drawn to.

Acknowledgements

This research was supported by the European Commission through the projects SMERobotics (FP7-287787-IP) and JAMES (FP7-270435-STREP). We would like to thank Caixia Cai, Kai Klinker and Artur Grunau for helping with data collection.

References

[1] Gioia Ballin, Matteo Munaro, and Emanuele Menegatti. Human action recognition from rgb-d frames based on real-time 3d optical flow estimation. In *Biologically Inspired Cognitive Architectures 2012*, pages 65–74. Springer, 2013.

[2] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5, 2001.

[4] Ling Gan and Fu Chen. Human action recognition using apj3d and random forests. *Journal of Software*, 8(9):2238–2245, 2013.

[5] Brian Gleeson, Karon MacLean, Amir Haddadi, Elizabeth Croft, and Javier Alcazar. Gestures for industry: Intuitive human-robot communication from human observation. In *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction*, HRI '13, pages 349–356, Piscataway, NJ, USA, 2013. IEEE Press.

[6] Shian-Ru Ke, Le Uyen Thuc Hoang, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. A review on video-based human activity recognition. *Computers*, 2(2):88–131, 2013.

[7] Adam Kendon. *Gesture: Visible action as utterance*. Cambridge University Press, 2004.

[8] Hema Koppula and Ashutosh Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 792–800, 2013.

[9] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[10] Claus Lenz, Alice Sotzek, Thorsten Röder, Helmuth Radrich, Alois Knoll, Markus Huber, and Stefan Glasauer. Human workflow analysis using 3D occupancy grid hand tracking in a human-robot collaboration scenario. In *IROS*, pages 3375–3380. IEEE, 2011.

[11] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–14. IEEE, 2010.

[12] Andrew Maimone and Henry Fuchs. Reducing interference between multiple structured light depth sensors using motion. In Sabine Coquillart, Steven Feiner, and Kiyoshi Kiyokawa, editors, *2012 IEEE Virtual Reality, VR 2012, Costa Mesa, CA, USA, March 4-8, 2012*, pages 51–54. IEEE, 2012.

[13] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79(3):299–318, 2008.

[14] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25(1):24–38, 2014.

[15] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 716–723. IEEE, 2013.

[16] Georgios Th Papadopoulos, Apostolos Axenopoulos, and Petros Daras. Real-time skeleton-tracking-based human action recognition using kinect data. In *MultiMedia Modeling*, pages 473–483. Springer, 2014.

[17] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[18] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *ASSP Magazine*, pages 4–16, January 1986.

[19] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[20] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Human activity detection from RGBD images. *CoRR*, abs/1107.0169, 2011.

[21] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from RGBD images, February 14 2011. Comment: 2012 IEEE International Conference on Robotics and Automation (A preliminary version of this work was presented at AAAI workshop on Pattern, Activity and Intent Recognition, 2011).

[22] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[23] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297. IEEE, 2012.

[24] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, pages 379–385, 1992.

[25] Yu Zhu, Wenbin Chen, and Guodong Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 486–491. IEEE, 2013.