Multimodal Human Activity Recognition for Industrial Manufacturing Processes in Robotic Workcells

Alina Roitberg
Technische Universität München
Boltzmannstr. 3
85748 Garching
roitberg@in.tum.de

Nikhil Somani
fortiss GmbH
Guerickestr. 25
80805 München
somani@fortiss.org

Alexander Perzylo
fortiss GmbH
Guerickestr. 25
80805 München
perzylo@fortiss.org

Markus Rickert
fortiss GmbH
Guerickestr. 25
80805 München
rickert@fortiss.org

Alois Knoll
Technische Universität München
Boltzmannstr. 3
85748 Garching
knoll@in.tum.de

ABSTRACT

We present an approach for monitoring and interpreting human activities based on a novel multimodal vision-based interface, aiming at improving the efficiency of human-robot interaction (HRI) in industrial environments.

Multi-modality is an important concept in this design, where we combine inputs from several state-of-the-art sensors to provide a variety of information, e.g. skeleton and fingertip poses.

Based on typical industrial workflows, we derived multiple levels of human activity labels, including large-scale activities (e.g. assembly) and simpler sub-activities (e.g. hand gestures), creating a duration- and complexity-based hierarchy. We train supervised generative classifiers for each activity level and combine the output of this stage with a trained Hierarchical Hidden Markov Model (HHMM), which models not only the temporal aspects between the activities on the same level, but also the hierarchical relationships between the levels.

Categories and Subject Descriptors

I.5.2 [**Design Methodology**]: Classifier design and evaluation, Pattern analysis

General Terms

Theory, Design, Experimentation

Keywords

Human activity recognition; Hierarchical Hidden Markov Model; Industrial robotics; Cognitive robotics

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
ICMI 2015, November 9–13, 2015, Seattle, WA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3912-4/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2818346.2820738.

1. INTRODUCTION

Vision-based human activity recognition has a strong potential to improve the quality of human-robot-interaction, e.g., for communication, dynamic task adaptation, or safety. A lot of work has been done recently with a focus on recognizing activities based on human skeleton tracking. Industrial scenarios are typically more constrained and require higher precision, especially for manipulation centric scenarios where hand movement is a key element.

Different uses of human activity recognition may require different levels of granularity. While the hand movement might be important in manipulation tasks, the position of the human in the working area might be sufficient for other tasks. In order to handle these different requirements, we propose a multi layer representation of activities (Fig. 1), from coarse identification to fine grained detection.

Industrial usage requires a higher level of robustness. Different sensors suffer from their individual limitations. In addition, certain sensors are optimized for specific use cases, e.g., hand-, head-, or skeleton tracking (Fig. 2). Therefore, a combination of multiple different sensors is required to provide adequate information to the system and to ensure robust communication between the human and the robot.

We designed a set of typical industrial workflows and derived four levels of human activities, ranging from large-scale activities (e.g., assembly) to simple hand gestures, which span a duration- and complexity-based hierarchy. A major contribution of this work is the machine learning based approach for multi-level recognition of human activities with respect to the defined semantics of this activity hierarchy.

The evaluation of this system was conducted on a data set containing 98 recordings of six different people, four different interaction scenarios and 38 activity types, using manual frame-wise ground-truth labeling and 4-fold cross validation. We systematically compare and discuss different generative classifiers for the first recognition stage, the impact of dimensionality reduction and the effects of combining data from multiple sensors compared to using only one sensor. The best recognition results were achieved using multiple input modalities, and recognizing the human activity by combining a SVM with radial-basis-function kernel with a HHMM. The average recognition accuracy varied between 81 % and

98 % for different levels of abstraction, which is especially significant since it was evaluated across multiple different people. A video showcasing results from the activity recognition framework can be found online.[1]

2. RELATED WORK

Human activity recognition is a well-studied area, hitherto mostly focused on 2D video data [13, 17]. Over the past few years the methods for obtaining high-quality 3D data have drastically improved due to the emergence of new affordable RGB-D sensors such as Microsoft Kinect or Asus Xtion. This progress strongly influenced the activity recognition research, with a large fraction of recent works based on 3D skeleton data captured by these devices [11, 12]. In our work, we also use the Kinect v2 and Leap Motion sensors, which are not yet popular for human activity recognition.

Various machine learning methods have been used for human activity recognition, with a systematic overview of the existing approaches provided in [1, 13]. In general, the ways of addressing this problem can be divided in two categories. In the first category, each feature vector carries all the information used for prediction and is directly classified without considering the temporal aspect. Such discriminative algorithms are widely used for activity recognition, with SVM and KNN being the most prominent approaches [2, 4, 12, 19]. In contrast, generative models such as Hidden Markov Models (HMM) also handle the temporal relationships between different classes of activities. Yamato et al. [18] were the first to use discrete HMMs for human activity recognition. In recent years, discriminative and generative approaches were often combined (e.g. SVM-HMM) [9, 6]. Fine et. al [3] extended HMM to a Hierarchical Hidden Markov Model (HHMM), which is capable of modeling parent-child relationships between the states and has also been used for human activity recognition [10].

The Cornell Activity Dataset (CAD) [16] includes 12 activities from five different domestic environments, e.g., cooking or brushing teeth. Sung et al. [16] proposed to decompose complex activities into sub-activities and to use a maximum entropy Markov model for recognition. With this method the authors reached precision and recall values of 67.9 % and 55.5 % respectively, with leave-one-person-out cross validation. Koppula et al. [7] reported significantly improved recognition results (precision 80.8 %, recall 71.4 %) by using structured support vector machines on the same dataset. Since no public datasets are available for the scenario of human-robot cooperation in an industrial context, we used a database that we recorded ourselves.

Hand gestures as an intuitive way of communication during industrial assembly tasks have been recently discussed by Gleeson et al. [5]. The results of the user study conducted in this work has shown the high potential of body gestures as an efficient way of interaction, although the interpretation of the gestures by a user unfamiliar with the definitions is highly context sensitive. These findings demonstrate the importance of realizing HRI systems based on body gestures, which is also the focus of our work. Despite the findings of Gleeson et al., research on human activity recognition addressing industrial environments has been sparse so far.

The work most similar to ours is that of Lenz et al. [8],

[1] https://www.youtube.com/watch?v=ggb6nUOEcjE

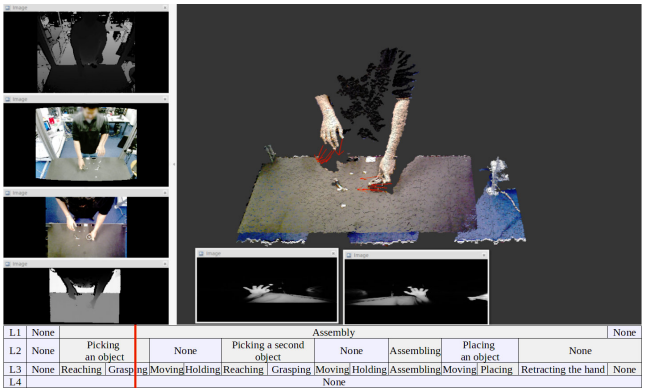


Figure 1: HRI workcell used for human activity recognition in industrial assembly scenarios. Data from the Asus Xtion Pro sensor is processed into a 3D point cloud and further extended with the finger tip positions obtained by Leap Motion sensors (red arrows). The timeline bar on the bottom shows the suggested multilevel annotation of human activities.

where the authors automatically recognized the activities of humans who were sitting in front of a work table and collaborating with an industrial robot. They tracked the positions of the humans' hands using 3D occupancy grids and used a composite HMM for the recognition of hand gestures and workflow analysis.

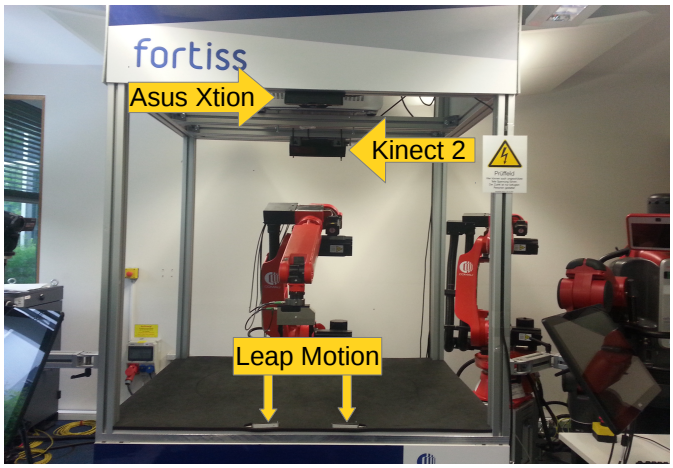
In our previous work, we used skeleton information provided by a Microsoft Kinect sensor with a combination of Random Forests and HMM for the recognition of simple human activities in industrial contexts [15]. In this work, we exploit multiple and more advanced modalities to recognize an extended set of more complex industrial activities with different levels of granularity. Similar to the previous work, a discriminative classifier is combined with a temporal model. However, the key component of the proposed recognition framework, i.e. its hierarchical modeling with HHMM, is a new aspect.

3. MULTIMODAL INTERFACE DESIGN

We specifically target production processes in robotic workcells. While typical domestic activities (e.g. jumping, waving) are well represented using the body skeleton, industrial systems have to accurately handle activities centered around the hands. Thus, industrial HRI-interfaces additionally require accurate articulated hand tracking. Further challenges of such systems are complex and occluded environments, and visibility limitations caused by compact sizes of workcells.

3.1 System Setup

We propose a new multimodal HRI-system, that is integrated in a typical industrial workcell and equipped with three different kinds of sensors (Fig. 2). A Microsoft Kinect v2 RGB-D sensor is mounted in front of the person, facing diagonally from above. The Kinect v2 is used to detect and track the human skeleton. An Asus Xtion Pro is placed on the metal cage facing straight down towards the table, and is intended for detecting objects on the table. For precise tracking of the hand skeleton, the Leap Motion Sensor is used. It is capable of calculating the position, orientation,



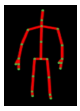
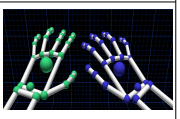
Property	Kinect v2	Asus Xtion PRO	LeapMotion
Hardware basics	Infrared (IR) light source, RGB and IR-cameras, microphone	Infrared (IR) light source, RGB and IR-cameras, microphone	Three IR LEDs , two IR cameras
Raw image data and resolution	RGB: 1920×1080 Depth: 512x424	RGB: 1280x1024 Depth: 640x480	Both IR cameras: 640x240
Data frequency	30 FPS (15-20 in our system)	30 FPS	20 to 200 (around 90 in our system)
Input for the proposed HRI system	Skeleton tracking with Kinect SDK v2: pose of 25 skeletal joints	Intended be used for object tracking from RGB-D pointclouds	Hand skeleton tracking with Leap Motion SDK: pose and velocity of hands and fingers
Data example			

Figure 2: Overview of the input modalities and their integration in the industrial environment. The workcell was equipped with three different types of sensors to capture human activities. A Kinect 2 sensor is frontally facing the worker. An Asus Xtion Pro sensor monitors the work table (for future object tracking). Leap Motion sensors are integrated into the tabletop.

direction and velocity of the finger tips with sub-millimeter accuracy. Due to the small size, accurate hand tracking can only be achieved within a very limited range. Hence, we have integrated two Leap Motion sensors inside the tabletop facing towards the expected working area of each hand.

3.2 Sensor Data Synchronization

When combining data from sensors produced by different manufacturers, data fusion quickly becomes a problem. Besides, the skeleton tracking software Kinect SDK v2 is restricted to Windows 8.1 only and multiple Leap Motion sensors cannot be connected to the same machine due to the lack of driver support.

We implemented a communication and control framework which unifies the sensor data over multiple machines and operating systems using the Robot Operating System (ROS) and ZeroC ICE. The master server manages the entire input as ROS messages, which can be easily saved and replayed as ROS BAG files. Limitations of the network bandwidth as well as the high resolution of the streamed videos occasionally lead to the loss of Kinect v2 data, which is subsequently corrected with a Kalman filter. The framework (Fig. 3) synchronizes the data at 30 frames per second and keeps track of the different coordinate systems using the ROS-TF library and manually calculated calibration data.

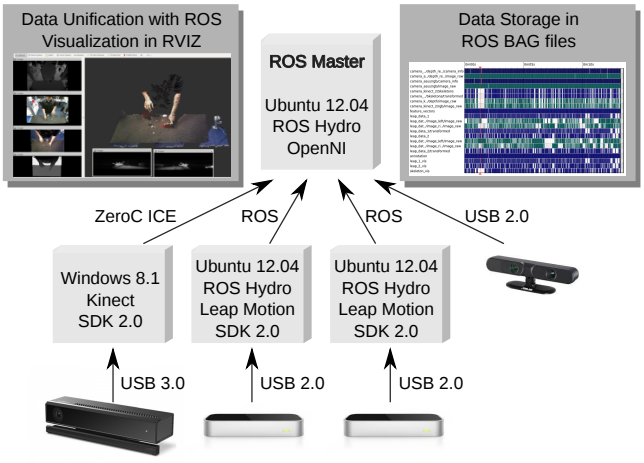


Figure 3: System overview: communication and control of multiple sensors.

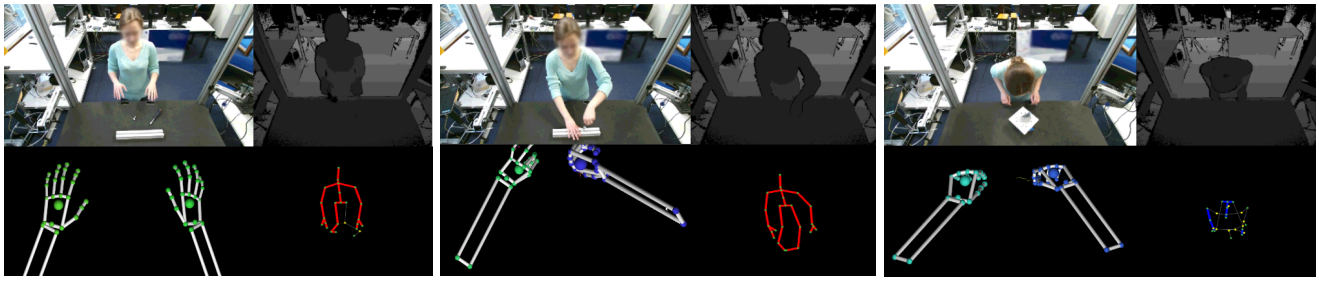
4. HUMAN ACTIVITY RECOGNITION

The pipeline for human activity recognition consists of several steps. In an offline stage, the scenarios and target activities at different levels are defined. This is followed by the design of the multimodal sensor setup and calibration. During runtime, the sensor data is processed to generate feature vectors that are used by the machine learning framework to estimate the activity labels.

4.1 Scenarios and Activities

We defined four different interaction scenarios for human-robot cooperation that are typically seen in industrial manufacturing processes. In the first scenario, a worker is fixing a screw on a work piece by hand and tightening it further using a tool. The second and third scenarios aim at the specification of welding tasks, by defining start and end points (second scenario) or a trajectory (third scenario) on a piece of metal using a predefined set of hand gestures. After the gesture based command is given, the welding task is assumed to be performed and the worker bends and takes a closer look at the result (Fig. 4). The fourth scenario is another assembly task with a different set of objects.

We analyzed the scenarios and derived a four-level hierarchy of activities (Fig. 5). The structure of the hierarchy is based on complexity, duration (which is highly correlated) and generalization. The activity of Level n (L_n) is linked by a set of grammar rules to sub-activities from L_{n+1}. In some cases, activities from L_n are generalizations of sub-activities from L_{n+1}, for example, the FIXING activity from L_2 is a generalization of FIXING WITH TOOL and FIXING WITH HANDS from L_3. While the set of activities for each



(a) None (b) Fixing (c) Leaning forward to check result

Figure 4: Visualization of multimodal input to the HRI interface. For each of the three exemplary activities, it depicts the RGB and Depth images (top left and top right) from a Kinect v2 sensor, the derived body skeleton (bottom right), and the hand skeletons (bottom left) detected by two Leap Motion sensors.

level was derived manually by observing the recordings, the exact semantics of the hierarchy was implicitly learned by the proposed classification algorithm.

4.2 Feature Calculation and Preprocessing

In this section, we describe our preprocessing framework for constructing feature representations from the sensor input. The inputs to our machine learning framework are poses of 25 skeleton joints calculated with the Kinect SDK v2 at a frequency of 15-20 FPS along with poses, velocities and directions of the hands and fingertips calculated with the Leap Motion SDK at approx. 90 FPS.

To estimate the current state of activity, meaningful features have to be carefully selected. Since the worker's legs are completely covered by the work table, skeleton information about the lower body joints has very low confidence and is hence ignored. We also calculate a set of position, orientation and motion features, that are suited for our application. Each position feature contains the 3D coordinates (X, Y, Z) transformed in the coordinate system of the body (skeleton joints) or the hand (finger tips). Orientation features are either angles with a single value (α) or 3D rotations represented as Euler angles $(roll, pitch, yaw)$.

Due to the dynamic nature of human activities, motion features carry significant information about human behaviour. Thus, we capture the change of feature values within certain time segments, i.e., the delta value as well as the variance within a segment.

Kinect v2 features. We use the positions of the hands, wrists and elbows and their motion information in the segment of the last 0.5 s. We put strong emphasis on the motion of the skeletal angles of the upper body joints as well as the orientation of the hip-, shoulder- and spine-line. Since the Kinect SDK does not provide any velocity information, we calculate the motion information in the last 0.5 s, 1 s as well as the motion history in the frame segments of 0.5 s to 1 s and 1 s to 2 s.

Leap Features. We use the positions of the fingertips and the hands, their orientations as Euler angles and their velocities, as provided by the Leap Motion SDK.

We use the same set of features for the training of L_1 to 3. Since L_4 consists of hand gestures intentionally performed with the right hand, we dismiss a set of features obviously linked to the left body part. After the selection, the feature set for L_1, L_2 and L_3 consist of 162 Kinect v2 features and 96 Leap Motion features, while the set for L_4 contains 146 Kinect v2 features and 45 Leap Features.

Since our feature vectors are derived from multiple sources observing overlapping parts of the same scene, they have a high number of redundant dimensions that also differ in their representative power for our application. Dimensions with low information content often result in noisy training data that adversely affects the classification results. Furthermore, high dimensionality requires higher computational power, which is also an important aspect since we aim for the development of a live interaction interface. We therefore use Principal Component Analysis (PCA) for dimensionality reduction.

4.3 Classification

Inferring human activity from skeleton information is a complex task with various facets. On the one hand, a single data frame with wisely selected pose features already provides a significant amount of information and can be categorized with a discriminative classifier, such as SVM or Random Forests. On the other hand, transitions between different activities have a strong causal and temporal aspect which can be modeled with generative methods as a part of the recognition approach, e.g., Moving an object requires Grasping an object as its predecessor. Furthermore, multiple abstraction levels for activities imply additional causal relationships inside the hierarchy. Integrating these dependencies in the recognition framework is crucial for two reasons. Firstly, knowing current activity states of the other levels would provide additional information for classification. Secondly, a strict hierarchical model ensures consistency of the classification output, e.g., L_3 activity Selecting trajectory and L_2 activity Assembling, can not occur simultaneously.

4.3.1 *Hidden Markov Models*

An excellent way to model such sequential processes are Hidden Markov Models (HMM), which are widely used in speech and video recognition. First introduced by Rabiner, a discrete HMM is given by an alphabet of n possible hidden states and a set of m possible observations [14]. Assuming there are n hidden states and m possible emissions, a 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 the initial probability vector of hidden states.

Algorithm 1 Training of a discriminative classifier

1: **Input:** $X = [X_1, ..., X_n]$ - a set of n feature vectors with m features; start and end of each recording segment is marked in X. $S^1 = [S_1^1, ...S_n^1]$ - annotation set containing ground-truth states of activities, with s possible states.
2: **Output:** C^1 - a discriminative classifier. Y^1 - a sequence of predicted states constructed by classifying X with C^1. Y^1 is later used as input for the second phase.
3: Randomization: from the given training data X, randomly select a subset ($\alpha = 75\,\%$) of samples (feature vectors) $X_{.75}$ and the corresponding annotation $S_{.75}^1$. This step is done to avoid over-fitting in the second training phase.
4: We use $X_{.75}$ and $S_{.75}^1$ for the training of a supervised classifier C^1 (e.g. SVM).
5: We use C^1 to categorize the whole initial training data set X (where $1-\alpha$ is the percentage not used for training of C^1), which results in the predicted sequence Y^1 with n samples.
Note: the notation a_c^b uses both, subscript c and superscript b as indices. The superscript does not represent the exponent of the variable. In most cases, the variable superscript b depicts a level in the activity hierarchy, while the subscript c is a simple enumeration index.

The classification problem is equivalent to the estimation of the most likely sequence of hidden states that took place with a previously trained model (π, A, B) and an observed sequence of emissions X. This problem can be solved with the Viterbi Algorithm with a run-time complexity of $O(n^2T)$, where n is the number of hidden states and T the length of the sequence.

For our activity recognition approach, we propose a strictly hierarchical two-stage machine learning framework, which combines discriminative and generative classification methods. The lowest level L_1 is not subordinate to any other levels and is processed independently from them. Consequently, the hierarchical aspect is not present at this stage and we use a discriminative classifier with a HMM for this stage (Algorithms 1 and 2). The other three activity levels are modeled using a Hierarchical Hidden Markov Model (HHMM).

4.3.2 Hierarchical HMMs

Given n levels of states (activities), the recognition algorithm is intended to produce n state predictions for each data frame, while taking into account the semantics of the learned hierarchy. In the previous section, we described our classification approach on the lowest level of the hierarchy, which is independent from the other levels, by combining a discriminative classifier with a HMM. In this section we describe the classification approach for all levels present in the hierarchy.

Using a hierarchical model is highly significant for preserving consistency between different levels, since the proposed structure of activities follows certain semantic rules. Fig. 5 shows the conditional probabilities of activities from the neighboring levels of abstraction. In other words, it is the probability $P(a_n|a_{n-1})$ of an activity a_n from a subordinate level L_n occurring simultaneously with an activity

Algorithm 2 Training of a HMM

1: **Input:** Y^1 - a sequence of predicted states from the first phase of training. $S^1 = [S_1^1, ...S_n^1]$ - annotation set containing ground-truth states of activities, with s possible states.
2: **Output of both phases:** $(C^1, (\pi, A, B)^1)$, a trained model for activity recognition on level 1, where C^1 is a discriminative classifier used in the first phase and $(\pi, A, B)^1)$ is the HMM of the second phase.
3: Calculate the transition matrix A $(s \times s)$ by calculating the transition probabilities for each pair of possible states from the annotation sequence S^1.
4: Calculate the emission matrix E $(s \times s)$ by calculating the probabilities for possible combination of states (given by S^1) and emissions (given by Y^1). The predictions of C^1 are used as observed emissions. The set of possible states and emissions is equivalent in our case.
5: Calculate the vector of initial state probabilities, π. π is calculated by counting the frequencies for each possible state of being the first state in a sub-segment.

Algorithm 3 Training approach with HHMM for Level K

1: **Input:** $X = [X_1, ..., X_n]$ - a set of n feature vectors with m features; start and end of each recording segment are marked in X. $S^k = [S_1^k, ...S_n^k]$ - annotation set containing n ground-truth states of activities for the current level K, with s^k possible states and s_i^k referring to the i-th possible state in the state set of level K. $S^{k-1} = [S_1^{k-1}, ...S_n^{k-1}]$ - annotation set containing n ground-truth states of activities for previous level $K-1$, with s^{k-1} possible states.
2: **Output:** The training of a higher level of a HHMM results in s^{k-1} separate trained classification models $(C^k_{s_i^{k-1}}, (\pi, A, B)^k_{s_i^{k-1}})$, where s_i^{k-1} is the state of the previous level, to which our model belongs.
3: Sorting the training data (X, S^k) by the state of previous level $K-1$: we divide the training data into s^{k-1}, where each sub-set corresponds to a possible state of previous level $K-1$. This results in a set of feature vectors and corresponding annotation of level K for every state of level $K-1$: $T = [(X^1, S^{k,1}), ..., (X^{s^{k-1}}, S^{k,s^{k-1}})]$.
4: Training of multiple models: For every possible state $s_i^{k-1}, i \subset [1, s^{k-1}]$, we train a classification model $(C^k_{s_i^{k-1}}, (\pi, A, B)^k_{s_i^{k-1}})$ with the same training approach used for the lowest activity level, as described in before.

a_{n-1} from a higher level of abstraction L_{n-1}. The conditional probability of 1 means that the subordinate activity is a specialization of a higher activity, while a value of 0 (a missing edge in Fig. 5) shows that those two activities (from different levels) never occur simultaneously in the database.

The Hierarchical Hidden Markov Model (HHMM) is an extension of HMM, that is capable of handling a topology of states based on parent-child relationships between multiple levels[3]. A HHMM consists of a set of simple HMMs with separate model parameters (π, A, B). The essence of a HHMM is a strict hierarchical order of the levels $L_1, ..., L_n$, where every possible state of the L_{k-1} itself contains an HMM for the classification of L_k. The algorithm for training a HHMM is presented in Algorithm 3.

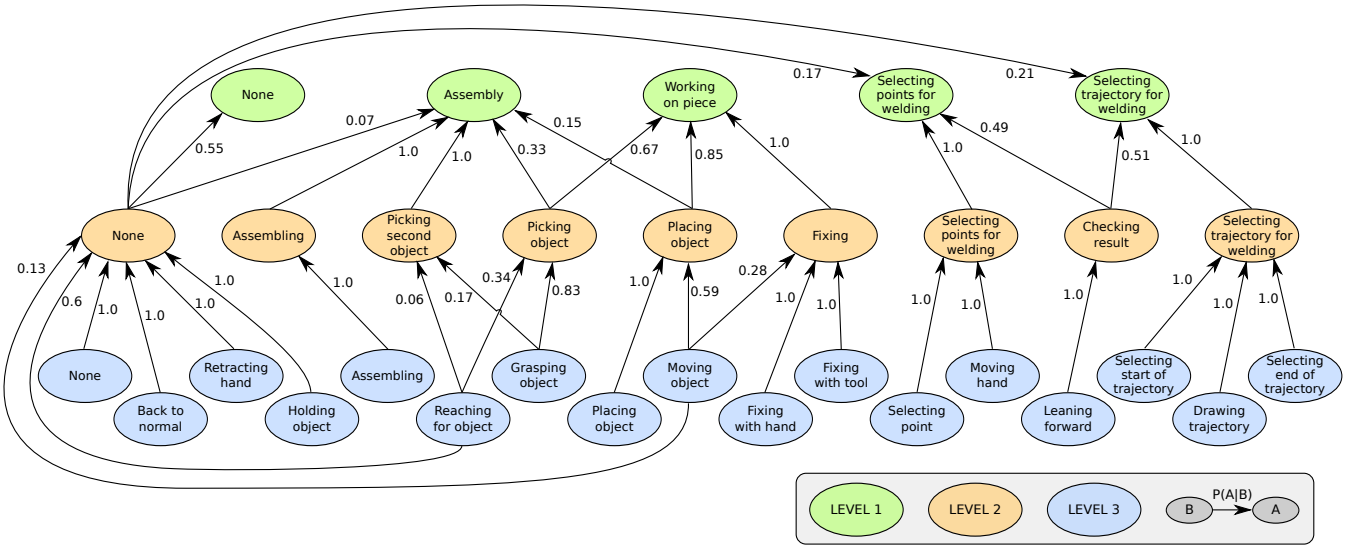


Figure 5: Conditional probabilities of the activities from levels L_1 to L_3 calculated from the collected database.

5. EVALUATION

The framework was evaluated on a database containing 98 recordings of four interaction scenarios with six different people. Video and skeleton data was recorded as ROS BAG files with manual frame-wise annotation added to it as a post-processing step. Please note that the data collected from the Asus Xtion sensor was not yet used for the activity recognition, but will be integrated in the system for object detection. The whole data set contains 33032 data frames after synchronization.

5.1 Methodology

Our approach was evaluated using 4-fold cross validation, where the data set was randomly divided into four equal subsets containing approx. 25 recordings each. It was ensured that each scenario was equally represented in each data set, while no such regularity was made as it came to different users. This was made intentionally, since recognition of unknown people is important for the practical usage of our system and the possibility of a person being present only in the test set or training set is a reasonable approximation.

Since each frame was annotated manually, accuracy of the ground-truth labeling itself should be taken into account. Due to the high frequency (30 Hz), slight annotation glitches during activity transitions are inevitable. The impact of this error is strongly dependent on the activity duration. For short dynamic activities, e.g., PUTTING THE THUMB DOWN, that start and end within a fraction of a second, false classification of a few frames at the beginning and the end can lead to a strong decline in accuracy.

For practical use, it is important that an activity is detected in its essential phase. To address this issue, we evaluate our system by counting correctly classified **activity sequences**, in contrast to correctly classified frames. We define a single activity sequence as the segment of frames during which the ground-truth activity label has not changed.

5.1.1 Metrics

The following notation is being used for the evaluation: $[A_1, ..., A_n]$ represent possible activity labels; A_i^{corr} is the number of correctly classified instances of class A_i; A_i^{total} is the true total number of instances of class A_i; A_i^{pred} is the number of instances classified as class A_i.

$$Avg.Accuracy = \frac{\sum_{i=1}^{n} \frac{A_i^{corr}}{A_i^{total}}}{n} \quad (1)$$

$$P = \frac{A_i^{corr}}{A_i^{pred}}, R = \frac{A_i^{corr}}{A_i^{total}}, F1 = 2 \times \frac{P \times R}{P + R} \quad (2)$$

In order to avoid a bias towards overrepresented classes, we use the average accuracy over all possible activity classes as the evaluation metric for the comparison of different classifiers (Fig. 6). In Table 1 we present a detailed evaluation of the recognition framework for each activity individually and evaluate the performance when using single sensor and multi sensor input. We calculate the Precision (P), Recall (R) and F1-score (2), with the F1-score being the key metric for single sensor and multi sensor comparisons.

5.2 Recognition and Prediction Results

We have tested several widely used classification algorithms for the first phase of the training and achieved best recognition results with a RBF-SVM estimator (98 %, 92 %, 81 %, 81 %), closely followed by linear SVM. Good classification results were also achieved by using Random Forests or K-Nearest-Neighbors with 20 neighbors (KNN-20), while KNN-200 and Gaussian Naive Bayes performed poorly. Detailed evaluation results are shown in Fig. 6.

Table 1 shows classification results for each activity individually and compares different subsets of features (*Leap*, *Kinect v2* and *Kinect v2 + Leap*), while Figure 7 shows the confusion matrices for every activity level with the best feature set (*Kinect v2 + Leap* for L_1, L_2, L_3 and *Leap* for L_4).

The lowest level $L1$ consists of a small number of large-scale activities and had the best recognition results, with 99 % of the sequences classified correctly when using *Kinect v2 + Leap* features. The confusion between the activities SELECTING POINTS FOR WELDING and SELECTING TRAJEC-

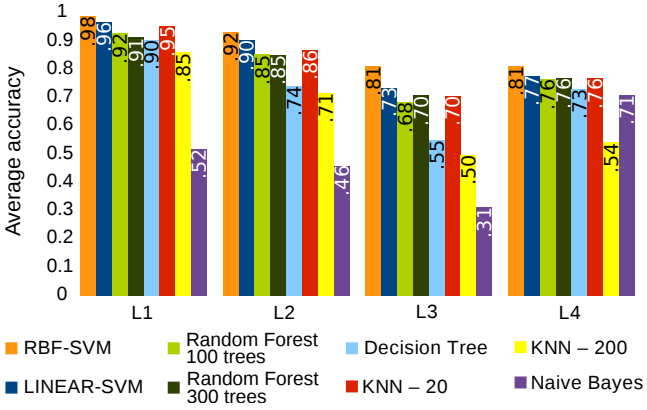


Figure 6: Average classification accuracy of different discriminative classification methods considered for the first phase of training. Best results were achieved with RBF-SVM for all levels.

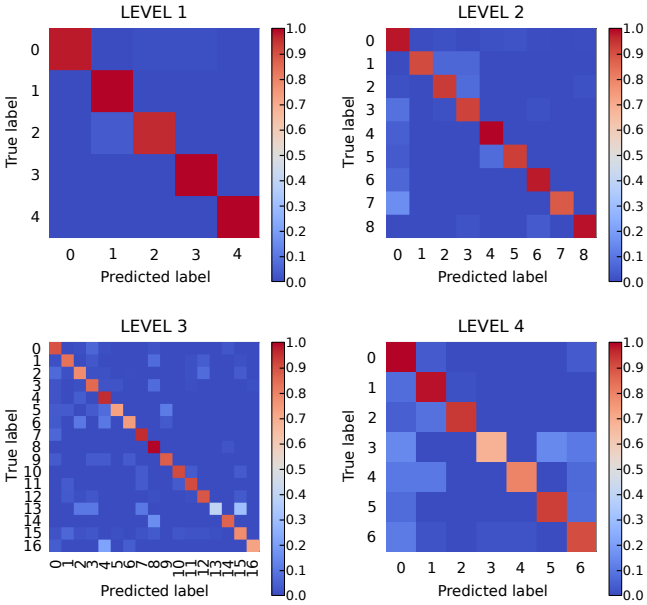


Figure 7: Confusion matrices for each level of activities with the best recognition approach (RBF-SVM + HHMM). Correspondence of the IDs to the actual activity names can be found in Table 1

		Kinect 2 + Leap			Leap only			Kinect 2 only		
		R	P	F1	R	P	F1	R	P	F1
ID	LEVEL 1									
0	None	0.98	1.00	**0.99**	0.91	0.98	0.94	0.87	0.99	0.92
1	Selecting points for welding	1.00	0.96	**0.98**	0.96	1.00	**0.98**	0.79	0.79	0.79
2	Selecting trajectory for welding	0.96	1.00	**0.98**	0.93	1.00	0.96	0.75	0.99	0.85
3	Working on a piece	1.00	1.00	**1.00**	1.00	1.00	**1.00**	1.00	1.00	**1.00**
4	Assembly	1.00	1.00	**1.00**	0.94	0.94	0.94	0.88	0.88	0.88
	Average	0.99	0.99	**0.99**	0.95	0.98	0.96	0.86	0.93	0.89
	LEVEL 2									
0	None	0.94	0.95	**0.94**	0.91	0.98	**0.94**	0.83	0.84	0.83
1	Picking a second object	0.88	1.00	**0.93**	0.75	0.80	0.77	0.81	0.93	0.87
2	Picking an object	0.90	0.98	**0.94**	0.83	0.95	0.89	0.68	0.94	0.79
3	Placing an object	0.89	1.00	**0.94**	0.79	0.99	0.88	0.75	0.98	0.85
4	Selecting points for welding	0.96	0.92	0.94	1.00	0.96	**0.98**	0.50	0.74	0.60
5	Selecting trajectory for welding	0.89	1.00	0.94	0.93	1.00	**0.96**	0.61	0.98	0.75
6	Assembling	0.94	0.79	**0.86**	0.94	0.71	0.81	0.50	0.73	0.59
7	Checking the result	0.85	1.00	**0.92**	0.28	0.76	0.41	0.85	0.85	0.85
8	Fixing	0.95	0.96	**0.95**	0.91	0.93	0.92	0.98	0.86	0.92
	Average	0.91	0.96	**0.93**	0.82	0.90	0.84	0.72	0.87	0.78
	LEVEL 3									
0	None	0.89	0.90	**0.90**	0.85	0.86	0.85	0.78	0.65	0.71
1	Placing an object	0.71	0.96	**0.81**	0.67	0.90	0.77	0.74	0.91	**0.81**
2	Reaching for an object	0.69	0.88	0.77	0.68	0.96	**0.80**	0.43	0.91	0.59
3	Retracting the hand	0.67	0.93	0.78	0.77	0.95	**0.85**	0.60	0.92	0.72
4	Selecting a point	0.88	0.95	**0.91**	0.94	0.88	**0.91**	0.50	0.75	0.60
5	Selecting the trajectory end	0.69	1.00	0.82	0.71	1.00	**0.83**	0.11	0.98	0.19
6	Selecting the trajectory start	0.64	1.00	0.78	0.79	1.00	**0.88**	0.34	0.98	0.51
7	Assembling	0.89	0.88	**0.89**	0.94	0.56	0.70	0.56	0.38	0.45
8	Back to normal	0.88	0.50	0.64	0.33	0.60	0.42	0.80	0.56	**0.66**
9	Drawing the trajectory	0.84	0.88	0.86	0.89	0.93	**0.91**	0.50	0.29	0.36
10	Fixing with a tool	0.89	0.88	**0.88**	0.96	0.75	0.84	0.96	0.59	0.73
11	Fixing with hands	0.84	0.89	**0.86**	0.79	0.67	0.72	0.82	0.72	0.77
12	Grasping an object	0.79	0.78	**0.79**	0.80	0.79	**0.79**	0.64	0.65	0.65
13	Holding an object	0.19	0.57	0.29	0.20	0.50	0.29	0.40	0.33	**0.36**
14	Leaning forward	0.73	0.96	**0.83**	0.35	0.57	0.43	0.85	0.81	0.83
15	Moving an object	0.64	0.87	0.74	0.76	0.82	**0.79**	0.57	0.86	0.68
16	Moving the hand	0.54	0.80	0.65	0.71	0.85	**0.77**	0.13	0.75	0.21
	Average	0.73	0.86	**0.78**	0.71	0.80	0.74	0.57	0.71	0.58
	LEVEL 4 (Gestures)									
0	None	0.95	0.58	0.72	0.93	0.70	**0.80**	0.98	0.23	0.37
1	Pointing	0.82	0.82	0.82	0.92	0.89	**0.90**	0.15	0.48	0.23
2	Thumb double-click	0.71	0.83	0.76	0.88	1.00	**0.93**	0.10	0.42	0.17
3	Two fingers: putting thumb down	0.30	1.00	0.46	0.63	0.95	**0.76**	0.00	nan	nan
4	Two fingers: releasing the thumb	0.42	1.00	0.59	0.74	0.96	**0.84**	0.06	1.00	0.12
5	Two fingers: thumb down	0.83	0.91	0.87	0.87	0.99	**0.93**	0.20	0.97	0.33
6	Two fingers: thumb up	0.75	0.99	0.86	0.84	0.99	**0.91**	0.28	0.98	0.44
7	Average	0.68	0.88	0.73	0.83	0.93	**0.87**	0.25	0.68	0.28

Table 1: Single sensor and multi sensor recognition results: Precision (P), Recall (R) and F1-score for each type of activity. Using multiple sensors shows the best average recognition results for L_1, L_2 and L_3. L_4 contains hand gestures, with the best classification results therefore obtained with the Leap Motion features only.

TORY FOR WELDING is to be expected, since the scenarios differ only in the way of specifying the trajectory.

L_2 represents decomposition of L_1 in more specific sub-activities. The best results were achieved when combining both sensors (avg. F1=0.93), except for the activities SELECTING POINTS FOR WELDING and SELECTING TRAJECTORY FOR WELDING, where best results were achieved when using *Leap* features only. This is because these scenarios contain predefined hand gestures, for which precise finger tracking is crucial, while rough body pose is insignificant. Hence, these activities are the two most poorly recognized labels when using *Kinect v2* data only. On the other hand, the activity CHECKING THE RESULT, where the human leans forward and takes a closer look at the workpiece, relies on the body pose only and can not be recognized with *Leap* data only (R=28 %). When using both sensors or *Kinect v2* only, the situation drastically improves (R=85 %). Combining both sensors still provides the best average recognition results.

L_3 has 17 sub-activities, with the best recognition results achieved with the combination of sensors (avg. F1=0.78). Since we aimed at a very precise set of sub-activities, the frequency of occurrence in the data set may vary.

In contrast to the other levels, best recognition results for L_4 were obtained by using Leap features only (F1=0.87). The activities at this level contain both very short dynamic gestures that last for fractions of a second (e.g. putting the thumb up or releasing it back down) and static gestures, which usually last longer. Short dynamic gestures were harder to classify, with PUTTING THE THUMB DOWN showing the lowest classification result (F1=0.76), while all static gestures have reached an F1 score over 0.9.

6. CONCLUSION

The contributions of this paper are two-fold. Firstly, we designed a multimodal interface integrated in a robotic workcell for HRI, tailored to recognize human activities and gestures. In spite of the diverse data sources and the complex communication framework, the system is easy to maintain and extend due to its compatibility with ROS. Secondly, we proposed a two-stages machine learning approach for the classification of human activities on multiple levels of abstraction. We use PCA for dimensionality reduction and combine a RBF-SVM estimator with a HHMM. The comparison of the recognition results using single and multiple sensors shows that multi-modality is crucial for our application. While the Leap Motion sensor offers the key data for the recognition of hand-centered activities, the Kinect v2 sensor provides better information about large-scale body movements, such as LEANING FORWARD. All in all, fusing multiple sensors is highly beneficial as it comes to recognizing both, hand-centered and body-centered movements. However, if the set of activities is limited to a certain type of movements, it might be useful to select the best data source and hence avoid unnecessary noise. The hand tracking provided by the Leap Motion sensor is therefore more significant for industrial manufacturing processes, than the body skeleton obtained with the Kinect v2 sensor. Besides higher success rates during the activity recognition, a combination of multiple data sources improves the system reliability by enlarging the observable area and adding input redundancy. Hence, a failure of one sensor can be compensated by another one. In the live system, it would be recommended to use three trained models, one for a combined feature set and one for each feature set originated from one of the sensors only. The system could use the combined model as long as all of the devices are providing reliable data, and switch to a single-sensor mode in case of a sensor drop out.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287787 in the project SMErobotics.

7. REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

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

[3] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

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

[5] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. 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, pages 349–356. IEEE Press, 2013.

[6] V. Kellokumpu, M. Pietikäinen, and J. Heikkilä. Human activity recognition using sequences of postures. In *IAPR Conference on Machine Vision Applications*, pages 570–573, 2005.

[7] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[8] C. Lenz, A. Sotzek, T. Röder, H. Radrich, A. Knoll, M. Huber, and S. Glasauer. Human workflow analysis using 3D occupancy grid hand tracking in a human-robot collaboration scenario. In *IROS*, pages 3375–3380. IEEE, 2011.

[9] B. Liang and L. Zheng. Multi-modal gesture recognition using skeletal joints and motion trail model. In *Computer Vision-ECCV Workshops*, pages 623–638. Springer, 2014.

[10] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on*, volume 2, pages 955–960. IEEE, 2005.

[11] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. 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.

[12] G. T. Papadopoulos, A. Axenopoulos, and P. Daras. Real-time skeleton-tracking-based human action recognition using kinect data. In *MultiMedia Modeling*, pages 473–483. Springer, 2014.

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

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

[15] A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert, and A. Knoll. Human activity recognition in the context of industrial human-robot interaction. In *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, pages 1–10. IEEE, 2014.

[16] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from RGBD images. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[17] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

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

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