Undefined 21-4 (2010) 1
IOS Press

Towards Automated Models of Activities of Daily Life

Michael Beetz *, Moritz Tenorth, Dominik Jain, and Jan Bandouch
Intelligent Autonomous Systems Group, Technische Universität München

Abstract. We propose *automated probabilistic models of everyday activities (AM-EvA)* as a novel technical means for the perception, interpretation, and analysis of everyday manipulation tasks and activities of daily life. AM-EvAs are detailed, comprehensive models describing human actions at various levels of abstraction from raw poses and trajectories to motions, actions and activities. They integrate several kinds of action models in a common, knowledge-based framework to combine observations of human activities with a-priori knowledge about actions. AM-EvAs enable robots and technical systems to analyze actions in the complete situation and activity context. They make the classification and assessment of actions and situations objective and can justify the probabilistic interpretation with respect to the activities the concepts have been learned from. AM-EvAs allow to analyze and compare the way humans perform actions which can help with autonomy assessment and diagnosis. We describe in this paper the concept and implementation of the AM-EvA system and show example results from the observation and analysis of table-setting episodes.

Keywords: Activity Modeling, Knowledge-based Action Analysis, Human Motion Tracking

1. Introduction

Our ultimate goal is to develop models of human everyday manipulation activities and represent these models as knowledge bases for different kinds of learning, reasoning, and analysis mechanisms. Having informative models of human activities enables machines to observe, analyze and compare human activities. Comparing the observations against models of "normal" behavior can help detect impairments. Comparing against previous observations of the same person allows to assess changes in the behavior. Also, powerful predictive and analytical models of human everyday activities are an important source of knowledge for assistive environments. We further believe that a lot of common-sense knowledge can be automatically derived from the respective activity models. For being able to detect changes both in terms of atypical motions (e.g. due to physical impairments) and at the level of activities (e.g. forgotten actions due to dementia), the models should cover a wide range of levels of descriptions, from very detailed motions to actions and activities.

We take as our running example table setting activities recorded in a kitchen environment, which are depicted in Figure 1. The upper figure contains the complete trajectory data for the right hand from five table setting episodes carried out by two different subjects. The three smaller images below show the subtrajectories for reaching for the cupboard handle, for taking objects from the table, and for reaching into a cupboard. The stereotypicality of the trajectories is quite surprising considering how many decisions need to be taken for carrying out the activities, including where to stand, how to reach, how to grasp, how to lift, where to hold the objects, etc, and and also considering the context-dependence of these decisions. This stereotypicality indicates that there is indeed a significant structure in human activities that can be learned and recognized by probabilistic models.

In this paper we describe a class of sensor-equipped software systems that we call *Automated Models of Everyday Activities* — AM-EvAs. AM-EvAs consist of automated activity observation systems, interpretation and abstraction mechanisms for behavior and activity

*Corresponding Author: Michael Beetz Intelligent Autonomous Systems Group, Department of Informatics, Technische Universität München. Boltzmannstr. 3, D-85748 Garching. E-mail: beetz@cs.tum.edu

*Corresponding Author: Michael Beetz Intelligent Autonomous Systems Group, Department of Informatics, Technische Universität München. Boltzmannstr. 3, D-85748 Garching. E-mail: beetz@cs.tum.edu

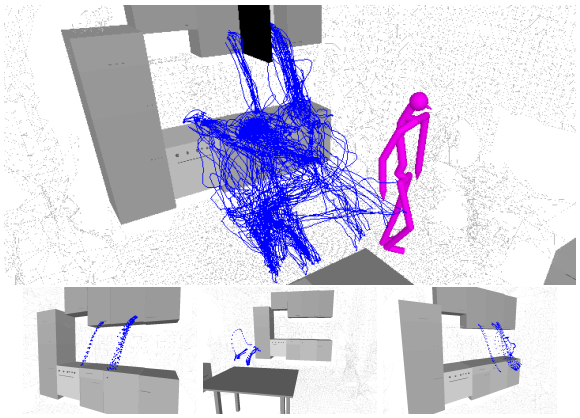

Fig. 1. Hand trajectory data recorded in table setting episodes. The upper picture shows the trajectory of the right hand during five table-setting episodes performed by two different persons. The lower pictures show segmented sub-trajectories for certain primitive actions.

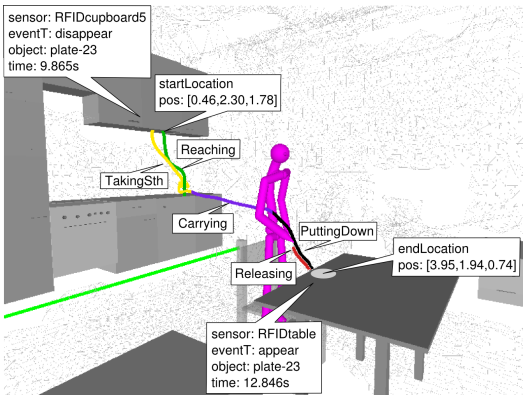

Fig. 3. Multi-modal observation of human activities. This figure illustrates which information the system provides and how it is grounded in the observations, namely the motion capture data, information from the sensor network like RFID detections, and inferred information like the *toLocation* of an action as the hand pose during the transition between a *PuttingDown* and *Releasing* motion.

data, a knowledge representation and reasoning system for symbolically representing the activity data, and a query system that allows AM-EvAs to answer semantic queries about the observed activities.

AM-EvAs are usually acquired by the following procedure, illustrated in Figure 2: Researchers in cognitive psychology plan experiments involving everyday manipulation activities. The activities are observed using a camera-based full-body motion tracking system and a sensor network including RFID (Radio Frequency IDentification) tag readers in cupboards and underneath the table, and magnetic sensors detecting whether doors are opened and closed. Figure 3 shows how the information in AM-EvA, including a sequence of poses, segmented and classified trajectories, observed events like an object appearing on the table, and the locations from which and to which objects are transported, is grounded in observations.

The sensor data stream is segmented and classified by learned classifiers that can recognize movement primitives such as reaching for an upper cupboard, picking up an object, etc. In the next step, these behavior data, segmented into movement primitives, are interpreted in order to compute more abstract macro-actions from the continuous data stream. These macro-actions are represented in a first-order temporal logic language which includes time intervals, events and actions as its basic concepts. Sets of these episode representations can then be used to compute joint probability distributions over the observed everyday activities which serve as a knowledge resource to answer key queries about these actions.

An important feature of AM-EvA is the knowledge-based framework that ties together the different modules. By representing the observations and all derived action descriptions together with their semantic meaning, AM-EvA can combine different pieces of information in an automated way and perform reasoning on the observed activities. The semantic description states, for example, that a certain number denotes the joint angle of a human elbow joint, or that an action segment is an instance of the action class *Reaching*. This information can be used to infer properties (e.g. that this motion has the goal to grasp an object), or to relate pieces of information (e.g. relations between the elbow joint and the shoulder joint) in a completely automated way. This part of AM-EvA is realized within the KNOWROB knowledge processing framework [11].

The key contributions of this paper are the following ones. We present an integrated system for observing, analyzing and interpreting complex human activities at different levels of abstraction. A knowledge-based framework integrates methods for human motion tracking, for learning continuous motion models, for motion segmentation and abstraction, and for probabilistic reasoning. At each level, all information in the system is represented in combination with its semantic meaning, which enables automated reasoning on the observations. The result are models that allow for an unprecedented depth of the automated analysis of human activities.

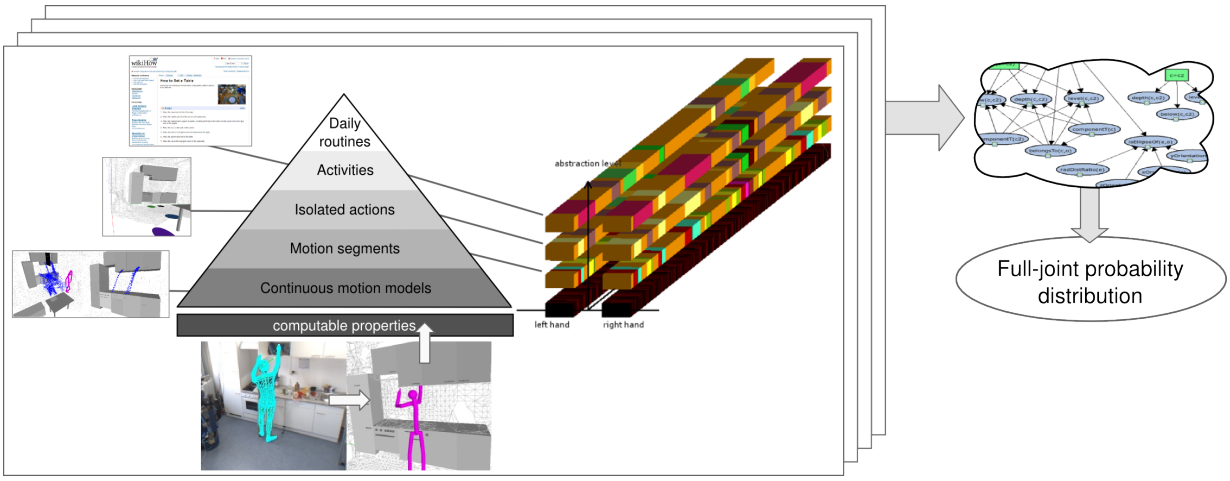

Fig. 2. Activity observation, interpretation, and analysis using AM-EvAs. Human actions are observed with a marker-less motion tracking system (lower part) and represented in a knowledge-based framework. Segmentation and abstraction methods generate more abstract action representations at several levels of abstraction from motions to actions and activities (upper left). The result can be used for learning statistical relational models of human activities (right part).

The remainder of this paper is organized as follows. We will first describe the different modules in AM-EvA, namely the human motion tracking system (Section 2), the techniques for learning continuous motion models (Section 3), for segmenting the stream of actions (Section 4), the methods for the symbolic action description and computation of more abstract representations (Section 5), and those for learning probabilistic models of complete activities (Section 6). We will then show some applications of the system (Section 7) and finish with our conclusions.

2. Observing Everyday Manipulation Activities

Observing human activities requires the estimation of human motions and associated poses at a detailed level. Often, commercial marker-based tracking systems are employed to make the estimation reliable. However, such systems are infeasible in real scenarios, as they are intrusive, expensive and difficult to set up.

We developed a marker-less motion capture system tailored towards application in everyday environments (Fig. 4). Our system comes with several improvements over both marker-based and state-of-the-art marker-less motion capture systems:

- Setup is fairly easy, cheap and unintrusive, requiring only the placement of several cameras in the environment. Three cameras are usually sufficient for tracking, although more cameras are recommended to account for occlusions from the environment.
- We derive a full body pose for each time step that is defined by an accurate 51 degrees-of-freedom articulated human model and corresponding estimated joint angles. This also enables us to calculate the trajectories of specific body parts, e.g. hand trajectories during a pick and place action.
- The system is functional without a preceding training phase and is unconstrained with respect to the types of motions that can be tracked.
- By incorporating their appearance, we are able to track subjects that act and interact in realistic everyday environments, e.g. by opening cupboards and performing pick and place tasks on objects.

Technically, our system estimates human poses in a recursive Bayesian framework using a variant of particle filtering. Each particle represents a human pose, given as a vector of joint angles. To account for the high dimensionality of the tracking problem, we developed a sampling strategy [2] that is a combination of partitioning of the parameter space [6] with a multi-layered search strategy derived from simulated annealing [4]. While the partitioning strategy enables us to take advantage of the hierarchical nature of the human body to reduce tracking complexity, the annealing strategy makes the system more robust to noisy measurements and allows us to use larger partitions at once that can be more accurately observed. This efficiently enables us to overcome local minima of the weight

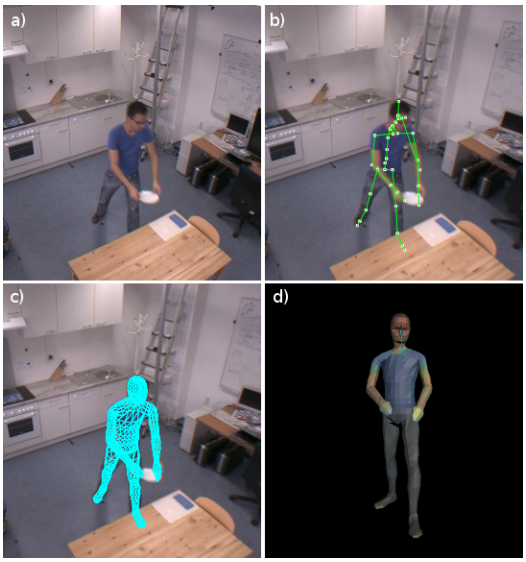

Fig. 4. Marker-less motion capture (one of four cameras): a) original image b) inner model c) outer model d) virtual 3D view with appearance model.

function and to gradually move particles towards the global maximum.

The observation model inside the particle filter framework is based on a comparison of 2D silhouettes extracted from multiple cameras (using common background subtraction techniques) with the corresponding model projections of the particles [1]. The motion model uses a constant pose assumption with body part dependent Gaussian noise added to account for the uncertainty. This makes it possible to track unconstrained human motions at comparably low frame rates (25 Hz). Inter-frame as well as intra-frame motion limits have been estimated using ergonomic expertise.

Our method is able to track subjects performing everyday manipulation tasks in realistic environments, e.g. picking up objects from inside a kitchen cupboard and placing them on a table (Fig. 4). Dynamic parts in the environment (such as objects being manipulated or opening doors) are filtered and ignored when evaluating particle weights based on a comparison between expected background and known foreground (human) appearance when evaluating particle weights. Occlusions from static objects in the environment (e.g. tables) are dealt with by providing blocked regions that will only be evaluated in areas that resemble the learnt foreground (human). As a rule of thumb for occlusion handling, every part of the human should be observable by (at least) three cameras to achieve good tracking accuracy. Therefore, areas with heavy occlusions

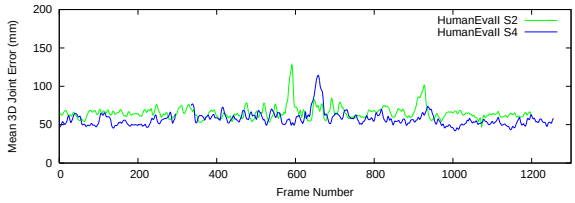

Fig. 5. Accuracy of marker-less motion capture as estimated using the *HumanEva II* benchmark [8].

should be covered by more cameras to gather sufficient information for successful pose estimation. We have validated our system on the *HumanEva II* test suite with available ground truth tracking data, and the results prove the validity of our approach (Fig. 5). The mean Euclidean joint position errors in 3D range around 5-6 cm (which might also be a systematic error due to differences in the human models used) and the tracking accuracy stays approximately constant throughout the two tested sequences. Note that the test suite provides a relatively simple scenario that does not contain occlusions, dynamic objects or manipulation tasks by the human subjects. Our system has also been successfully applied to more complex scenarios (see `http://memoman.cs.tum.edu` for videos).

The data we are using here have been released in the TUM Kitchen Data Set [10] and are publicly available for download[1]. The data set currently contains observations of 20 episodes of different kinds of setting a table, performed by four different subjects. It provides video data of four fixed, overhead cameras, motion capture data, as well as information from sensors embedded in the environment, e.g. RFID tag readers in cupboards and underneath the table, or magnetic sensors that detect if a cupboard door is being opened.

3. Continuous Motion Models

At the most detailed level, our models base their representations directly on the joint motions that are gathered by the marker-less full-body motion tracking system (together with information on object interactions from the sensor network in general). Even though the data at this level is very high-dimensional, it is reasonable to assume that it is nevertheless well-structured, because actions performed during housework are in many ways constrained (with respect to the expected limb motions, which are far from arbitrary)

[1] http://kitchendata.cs.tum.edu

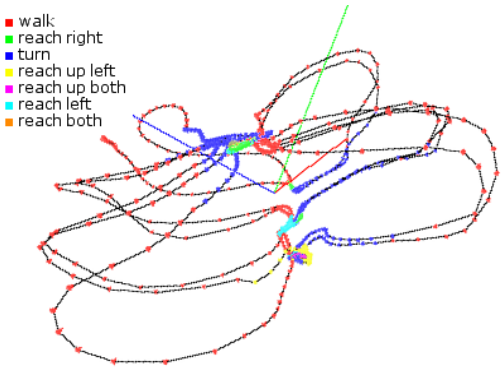

Fig. 6. GPDM with class constraints: low-dimensional embedding of an unconstrained pick-and-place task

and they follow clearly discernible patterns. These patterns can be made explicit by suitably embedding the high-dimensional data into a low-dimensional latent space.

Preliminary experiments with Gaussian process dynamical models (GPDMs) [14] indicate that the structure of embedded trajectories in the latent space may even serve as a starting point for the (unsupervised) segmentation of trajectories into meaningful fragments, whose sequential ordering in turn provides the input for a further interpretation of the overall action sequence in a (discrete) time-series model. Such an unsupervised approach will group motions mainly with respect to their kinematic or dynamic properties.

Usually, however, we do not want to relinquish control over the individual atomic actions that we consider, so it is advisable to directly incorporate the semantic labellings of action sequences as a further input dimension to the learning algorithms of, for example, GPDMs, and to consequently learn low-dimensional embeddings that seek to structure the latent space with respect to the labels. To this end, we extended learning algorithms for GPDMs with probabilistic constraints that ensure that points belonging to the same classes are close together while points belonging to different classes may be far apart, further structuring the latent space according to its semantic interpretation (see Figure 6). (A somewhat similar approach, which, focusing on specific inference tasks, considers a discriminative model, was proposed in [13].) Given a learnt mapping from the high-dimensional data space to the latent space, we can then perform classification of newly observed sequence data by maximizing the likelihood of the latent space configuration given the labels.

Since our models are generative, we can flexibly use them to either predict future motions that are likely to occur, evaluate the probability of an observed motion sequence (allowing us to detect peculiar actions/motions that are, for example, unusual given the overall actions that are supposed to be performed) or, as previously stated, infer the labels of a sequence, providing the discretized information for higher-level modeling. The labeling constitutes precisely the semantic interpretation that we require to analyze activities at higher levels of abstraction.

While GPDMs model the complete poses involved in an action, it is sometimes useful to describe just the motion of the relevant hand performing an action. This significantly lower-dimensional representation allows to distinguish different kinds of reaching trajectories, or to select trajectories for a robot to imitate. We are currently investigating the integration of the compact models for hand trajectories described in [9].

4. Action Segmentation

While the aforementioned approaches target at modeling trajectories and primitive movements, it is often sufficient to determine when one motion of a kind starts and ends, especially as input for higher-level processing. As a fast method for performing such a segmentation of actions we use linear-chain Conditional Random Fields (CRF). The CRFs are embedded into the knowledge processing framework and produce a sequence of instances of the respective motions, such as *Reaching* or *TakingSomething*, as result of the segmentation process.

The features used as input to the CRF are nominal and could be split into two groups: Pose-related features denote, for instance, if the human is extending or retracting the hands, or if the hands are expanded beyond a certain threshold. Information obtained from the environment model and the sensor network complements the pose-related features and states e.g. if the human carries an object, if a hand is near the handle of a cupboard, or if a drawer is being opened. These features are combined, and CRFs are learned on a labeled training set (Figure 7). The CRF-based segmentation is described in more detail in [10].

5. Hierarchical Symbolic Action Models

The result of the CRF-based segmentation serves as the input for higher-level action analyses. These are

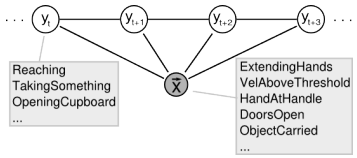

Fig. 7. Structure of the CRF used for segmenting human motions.

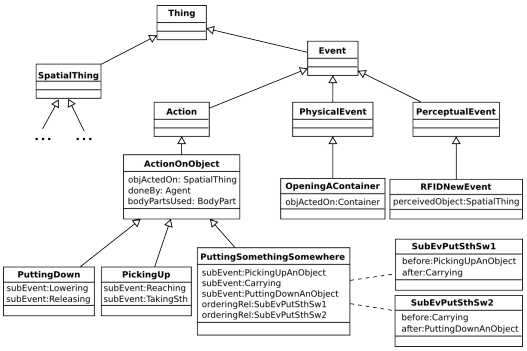

Fig. 8. Knowledge about classes of actions, events and objects that is represented in AM-EvA and used in the abstraction process.

performed on symbolic action descriptions, i.e. motion segments and actions which are formally represented in a knowledge base. AM-EvA builds upon the KNOWROB knowledge processing framework. Actions at all levels of abstraction are represented as instances of the respective action classes; their relations are abstractly specified at the class level.

Figure 8 shows a small excerpt of the AM-EvA ontology containing actions, including sub-actions and potential ordering constraints among these sub-actions, events, and spatial concepts like objects. The *subAction* relation can be used for abstracting from the observed sequence of motions to higher-level actions and activities, creating a hierarchical action model like the one shown in Figure 9.

Technically, the CRF based segmentation yields a sequence of motions (like *Reaching-14*, *TakingSomething-14*, etc) that are formally represented as instances of the respective motion classes (*Reaching-14* is an instance of the general class of *Reaching* motions, etc). Action parameters like the object that is manipulated, the start time, or the agent performing the action are abstractly described as first-order relations between the action and object instances. Which property an action has is determined automatically based on information from the sensor network: Observed events like RFID tag detections or registered door openings are related to simultaneously performed actions to determine e.g. the *objectActedOn* relation.

Starting from the sequence of low-level motions, the system generates more abstract action descriptions by applying transformation rules. In Figure 9, for example, it used general knowledge that actions of type *PickingUpAnObject* have sub-actions of types *Reaching* and *TakingSomething*. Whenever it finds such actions, it generates a new, higher-level action instance. Thereby, it propagates action properties from the lower levels of abstraction upwards, so that e.g. the *atLocation* of a pick-up event becomes the *fromLocation* of a transport action.

Exemplary results of hierarchical action models that were learned from sequences in the TUM Kitchen data (left, center) and the CMU MMAC data set [3] (right) can be found in Figure 10. They show how the short segments corresponding to low-level motions are combined to more and more abstract descriptions, e.g. to the segments of type *PuttingSomethingSomewhere* which are drawn in pink in the left picture. The center picture shows how the information about object interaction is propagated upwards in the abstraction hierarchy. In this picture, the color denotes which object is manipulated, and it can be seen how the object of lower-level actions for picking up and putting down an object is propagated upwards to the *PuttingSomethingSomewhere* action.

External sources of knowledge, e.g. textual descriptions, can easily be included to determine the type of activity that is being performed, or to compare the observations to the descriptions and spot differences. Examples of such external knowledge are step-by-step instructions from web sites like ehow.com, which we have successfully translated from natural language into a formal representation that is compatible to the knowledge in AM-EvA [12]. These converted instruc-

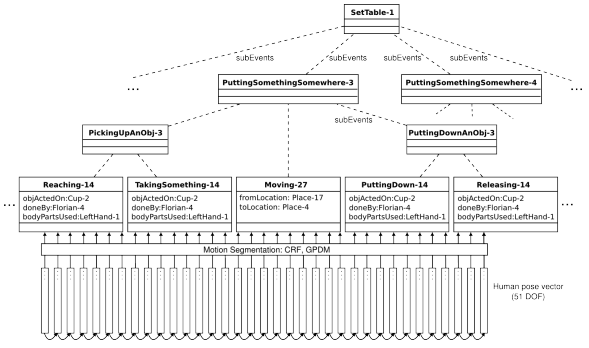

Fig. 9. Hierarchical action model constructed from observed human tracking data.

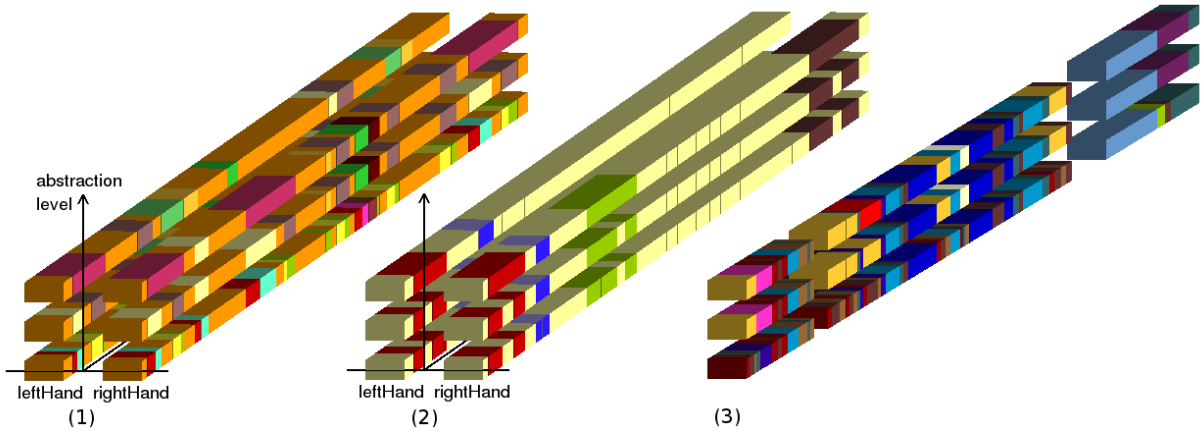

Fig. 10. Visualization of the action abstraction on the TUM Kitchen Data set and the CMU MMAC data set. (Left: TUM data, color based on the action type. Center: TUM data, color based on the manipulated object. Right: CMU data, color based on the action type. In the TUM data, there are separate stacks of sequences for the left and right hand, in the CMU data only one for both. Higher levels correspond to more abstract descriptions, the time dimension points towards the back.

tions describe tasks at the class level, while the hierarchical action models described here comprise instances of actions. For matching and comparing these, one needs to check which classes subsume which action instances. We will explore this direction of research in the near future.

6. Probabilistic Activity Models

The symbolic, relational data about actions we obtain through the application of hierarchical action models may serve as a source of information for the learning of probabilistic models, in which we seek to capture the uncertainty that permeates the domain of everyday activities. Different instances of an activity will typically feature a different set of actions with a different ordering and different parameterizations. Nevertheless, we can usually identify certain regularities and thus extract general principles from a body of training data and hope that these principles will indeed transfer to new instances of an activity that we may want to reason about. Reasoning tasks will usually need to consider a variable number of interrelated entities (such as actions and the objects these actions are applied to). Hence the relational character of the representation should be maintained. Statistical relational models are therefore the most appropriate choice for the representation of probabilistic action and activity models.

In particular, the AM-EvA framework supports two types of statistical relational models: Bayesian logic networks (BLNs) [5] and Markov logic networks [7]. Both formalisms essentially allow to represent a metamodel of probability distributions, i.e. a template for the construction of a concrete probability distribution that can be represented as a graphical model (a Bayesian network or Markov network). Though Markov logic networks are generally more expressive, applying them is, unfortunately, problematic in practice, since both learning and inference are typically harder – even if one restricts oneself to structurally simple models. We therefore use BLNs whenever possible.

A BLN is essentially a template for the construction of a mixed network, i.e. combination of a Bayesian network with a constraint network that filters out possible worlds that do not satisfy the constraints being represented. Formally, a BLN is a tuple $\langle \mathcal{D}, \mathcal{F}, \mathcal{L} \rangle$, where $\mathcal{D}$ is a set of declarations (of predicates/functions, entity types, entities and functional dependencies), $\mathcal{F}$ is a collection of conditional probability fragments, and $\mathcal{L}$ is a set of hard constraints in first-order logic. Given a concrete set of entities for which a probabilistic model is to be considered (e.g. a set of actions and objects), the BLN generates, in accordance with the set $\mathcal{D}$, a Bayesian network from $\mathcal{F}$ and a constraint network from $\mathcal{L}$. The resulting mixed network represents a full-joint probability distribution over the atomic sentences/random variables that apply to the entities for which the model is instantiated, providing a highly flexible representation that supports causal, diagnostic, inter-causal and mixed queries alike. In the

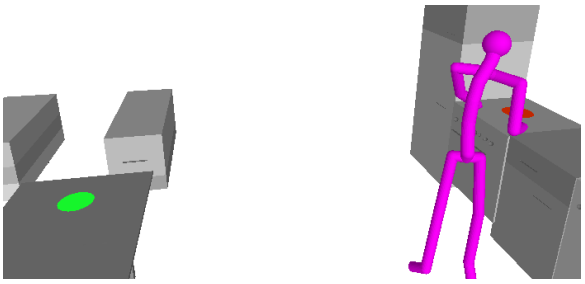

Fig. 11. Locations where a place mat is taken from (dark red) and put to (light green) during a table-setting task.

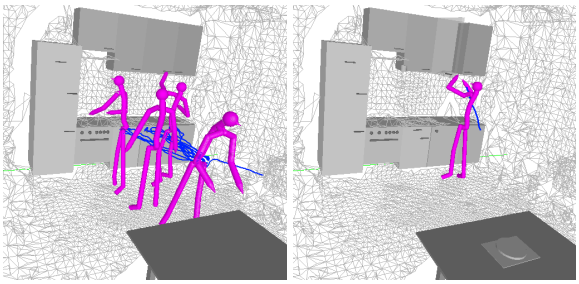

Fig. 12. Human pose sequences for setting a table (left) and taking a plate out of the cupboard (right).

following section, we will present some concrete applications of BLNs.

7. Applications

In this section, we will give an overview over the range of information AM-EvA provides — both in terms of logical knowledge and probabilistic knowledge — by presenting a number of queries to the system.

7.1. Logical/Relational Knowledge

We start with asking for everything that is known about the action segment *PickingUpAnObject100*:

```
?- owl_has('PuttingSomethingSomewhere100', Pred, O).
   Pred=type,              O=PuttingSomethingSomewhere ;
   Pred=subAction,         O=PickingUpAnObject150 ;
   Pred=subAction,         O=CarryingWhileLocomoting53 ;
   Pred=subAction,         O=PuttingDownAnObject151 ;
   Pred=objectActedOn,     O=placemat-1 ;
   Pred=doneBy,            O=florian ;
   Pred=bodyPartsUsed,     O=rightHand ;
   Pred=fromLocation,      O=loc(0.32,1.98,1.08) ;
   Pred=toLocation,        O=loc(3.2,2,0.74) ;
   Pred=startTime,         O=0.722562 ;
   Pred=endTime,           O=5.45968
```

AM-EvA has information about the type of the action, sub-actions it is composed of, the object that was manipulated, the agent who performed the action, and the locations and times where and when the action started and ended. This abstract information is grounded in the observed pose sequences, so the poses corresponding to actions and activities can be queried. For example, in the image on the left of Figure 12, we asked for the whole pose sequence of a table setting activity using the following query:

```
?- type(?Acty, 'SetTable'),
   postureForAction(?Acty, ?Posture),
   highlight_postures(?Posture).
```

Though the traces are only drawn for a single joint, the result includes the full human pose vectors for each point in time. The query depicted in the right image in Figure12 asks for all postures that are part of a *TakingSomething* motion, performed on a *DinnerPlate* in a *TableSetting* context:

```
?- type(?Acty, 'SetTable'),
   type(?Actn, 'TakingSomething'),
   subAction(?Actn, ?Acty),
   objectActedOn(?Actn, ?Obj),
   type(?Obj, 'DinnerPlate'),
   postureForAction(?Actn, ?Posture),
   highlight_postures(?Posture).
```

The predicate *trajForAction* extracts just the hand trajectories, which are a more compact and often more useful representation than the high-dimensional pose sequences. The trajectory for a *TakingSth* motion with a *fromLocation* on the table is obtained with the following query and visualized in Figure 1, bottom left:

```
?- type(A,'TakingSth'), fromLocation(A,From),
   on_Physical(From,T), type(T,'Table'),
   trajForAction(A,'rightHand',Traj).
```

The models also allow queries for action-related information, for example from which location to which location an object is transported (visualized in Figure 11):

```
?- type(A, 'PuttingSomethingSomewhere'),
   objectActedOn(A, O), instance_of(O, 'PlaceMat'),
   fromLocation(A, FL), highlight_location(FL),
   toLocation(A, FL),   highlight_location(TL).
```

We can also query for habits of a person, e.g. if cupboards are always opened with the left hand

```
?- forall(type(A, 'OpeningACupboard'),
   bodyPartsUsed(A, 'leftHand')).
Yes
```

The following query asks for objects that are carried with both hands, i.e. whether two simultane-

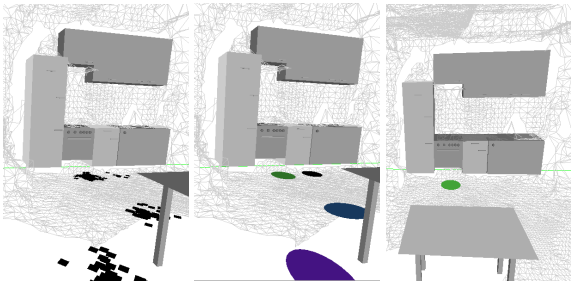

Fig. 13. Observed positions of manipulation actions (left), the positions clustered into places (center), and the result that has been learned as the "place for picking up pieces of tableware" (right). The pictures are visualized results of queries to the knowledge processing system.

ous *PuttingSomethingSomewhere* actions, performed by different hands, exist:

```
?- type(A1,'PuttingSomethingSomewhere'),
   type(A2,'PuttingSomethingSomewhere'),
   not(A1=A2),
   bodyPartsUsed(A1,'leftHand'),
   bodyPartsUsed(A2,'rightHand'),
   objectActedOn(A1,O),objectActedOn(A2,O),
   timeInterval(A1,I1),timeInterval(A2,I2),
   timeOverlap(I1,I2).

O = 'placemat-1'
```

7.2. Action-Related Concepts

In addition to querying for motion segments, the models can also be used for more complex reasoning. One example is to learn action-related features from the observations, again combining information from the motion tracking and the sensor network. Action-related concepts are all kinds of concepts that can be characterized by their role in actions, like for instance the place where a human is standing while performing certain activities, or the trajectories used for performing certain actions. Knowledge about these concepts can be used for recognizing intentions and for finding differences in how people perform the same action.

These concepts are autonomously learned from observations using the action models described in [11]. In a first step, the system retrieves the training data like the positions where the human was standing while performing manipulation actions (Figure 13 left) and the parameters of these actions. These positions are clustered with respect to the Euclidean distance, and the clusters are abstractly represented as places in the knowledge representation (Figure 13 center).

The system then learns a mapping between the actions, their parameters and the place where they were performed by training a classifier on the observed data. The resulting concepts effectively extend the class taxonomy with new classes like a "pick-cups-place" that are discovered in the observations. Figure 13 (right) shows the result of a query for a "place for picking up pieces of tableware".

7.3. Probabilistic Knowledge

We used parts of the logical knowledge that we derived from observations for training Bayesian logic networks. For instance, based on semantically interpreted data from the CMU-MMAC dataset [3], we trained a simple model of cooking activities. The training set contained 23 sequences of 16 subjects performing two types of activities, *BakingBrownies* and *CookingAnOmelette*. The BLN model, whose set of fragments is shown in Figure 14, relates cooking activities to the actions appearing within these activities, which are parameterized with a type, an agent and the object and location involved.

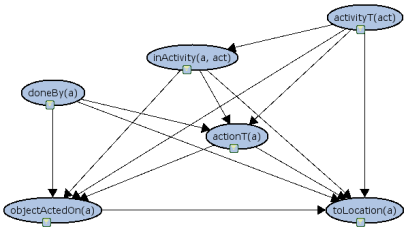

Fig. 14. BLN fragments of a simple model of cooking activities

As a very basic application of this model, we can infer the type of the activity given a number of observed individual actions,

$$
\begin{aligned}
&P(activityT(Act) \mid inActivity(A1,Act) = True \\
&\quad actionT(A1) = TakingSth \wedge objectActedOn(A1) = Bowl \wedge \\
&\quad inActivity(A2, Act) \wedge actionT(A2) = Cracking \wedge \\
&\quad objectActedOn(A2) = Egg\text{-}Chickens) \\
&\approx \langle\, BakingBrownies: 0.83,\ CookingAnOmelette: 0.17 \,\rangle
\end{aligned}
$$

Having observed that a bowl has been taken and that an egg has been cracked, the model favors *BakingBrownies*, because, even though the actions observed can appear in either activity, they play a more defining role in *BakingBrownies* (since the fraction of *Cracking* actions in *BakingBrownies* is, when compared to *CookingAnOmelette*, three times as high). Additional information will change our beliefs. If, for ex-

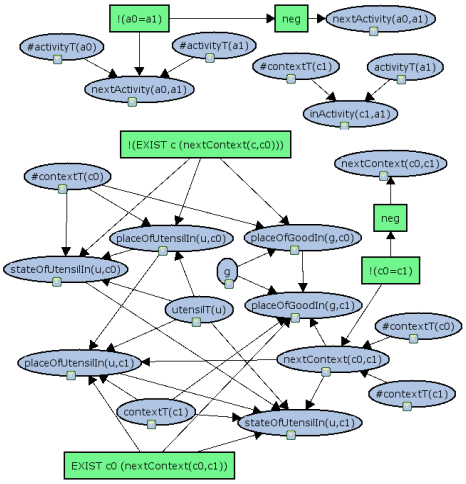

Fig. 15. Fragments in model of activity sequences. Rectangular nodes indicate preconditions for fragments to be applicable – used, for example to differentiate between activities at the beginning and in the middle of a sequence.

ample, a third action involved a frying pan being fetched, we could certainly identify the activity as *CookingAnOmelette*.

As a second example, consider the more elaborate fragment structure shown in Figure 15, a sequence model that considers the effect of activities on the places and states of objects. For instance, we could provide as evidence information on the locations or states of various objects in the environment and ask for the activities that are likely to have led to such a situation, and furthermore ask for the higher-level activity that these activities are all likely to be part of:

$$
\begin{aligned}
&P(\mathit{activityT}(A), \mathit{contextT}(C1), \mathit{contextT}(C2), \mathit{contextT}(C3) \mid \\
&\quad \mathit{inActivity}(C1, A) \wedge \mathit{inActivity}(C2, A) \wedge \mathit{inActivity}(C3, A) \wedge \\
&\quad \mathit{nextContext}(C1, C2) \wedge \mathit{nextContext}(C2, C3) \wedge \\
&\quad \mathit{utensilT}(U) = \mathit{Plate} \wedge \mathit{placeOfUtensilIn}(U, C3) = \mathit{Table} \wedge \\
&\quad \mathit{placeOfGoodIn}(\mathit{Omelette}, C3) = \mathit{Table}) \\
&\approx \langle\langle\ \mathit{Dinner}: 0.59, \mathit{Lunch}: 0.36, \mathit{Snack}: 0.04, \ldots\rangle, \\
&\quad \langle\ \mathit{CookingAnOmelette}: 0.52, \mathit{CookingNoodes}: 0.30, \ldots\rangle, \\
&\quad \langle\ \mathit{ServeFood}: 0.52, \mathit{SetTable}: 0.48, \ldots\rangle, \\
&\quad \langle\ \mathit{SetTable}: 0.52, \mathit{ServeFood}: 0.30, \mathit{Dining}: 0.18, \ldots\rangle\rangle
\end{aligned}
$$

Inversely, we could have provided information on a sequence of activities and asked for the most likely locations or states of objects these activities may have involved. With a full-joint probability distribution, either query is possible.

In the context of assistive technology, it is conceivable to specifically adapt such probabilistic models (or wrap a monitoring system around them) in order to enable a system to infer whether the (sequence of) actions it observed is typical with respect to the overall activity being carried out or whether the human subject has missed steps that were expected.

8. Conclusions

In this paper, we have described the concept and implementation of AM-EvAs, *automated models of everyday activities* for the perception, interpretation, and analysis of everyday manipulation tasks and activities of daily life. We have outlined the main components of AM-EvA, which are a full-body motion tracking system for people performing everyday manipulation tasks, various learning approaches that infer activity representations from the tracking data and segment continuous motions. Other system components combine these hybrid activity models with encyclopedic knowledge about everyday manipulation tasks and human living environments to provide prior knowledge for making learning and inference more tractable. AM-EvA seamlessly combines symbolic knowledge with observed behavior data structures through the use of relations and properties that are evaluated directly on AM-EvA's data structures and through data mining mechanisms that learn symbolic concepts from observed activity data.

We believe that AM-EvAs can help with the analysis of human actions to assess their level of independence and to diagnose potential impairments.

9. Acknowledgments

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems (CoTeSys)*, see also `www.cotesys.org`.

References

[1] J. Bandouch, F. Engstler, and M. Beetz. Accurate human motion capture using an ergonomics-based anthropometric human model. In *Proceedings of the Fifth International Conference on Articulated Motion and Deformable Objects (AMDO)*, 2008.

[2] J. Bandouch, F. Engstler, and M. Beetz. Evaluation of hierarchical sampling strategies in 3d human pose estimation. In *Proceedings of the 19th British Machine Vision Conference (BMVC)*, 2008.

[3] F. De la Torre, J. Hodgins, J. Montano, S. Valcarcel, and J. Macey. Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database. Technical report, CMU-RI-TR-08-22, Robotics Institute, Carnegie Mellon University, 2009.

[4] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision (IJCV)*, 61(2):185–205, 2005.

[5] D. Jain, S. Waldherr, and M. Beetz. Bayesian Logic Networks. Technical report, IAS Group, Fakultät für Informatik, Technische Universität München, 2009.

[6] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 3–19, London, UK, 2000. Springer-Verlag.

[7] M. Richardson and P. Domingos. Markov Logic Networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

[8] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, Brown University, 2006.

[9] F. Stulp, I. Kresse, A. Maldonado, F. Ruiz, A. Fedrizzi, and M. Beetz. Compact models of human reaching motions for robotic control in everyday manipulation tasks. In *Proceedings of the 8th International Conference on Development and Learning (ICDL).*, 2009.

[10] M. Tenorth, J. Bandouch, and M. Beetz. The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *IEEE Int. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS). In conjunction with ICCV2009*, 2009.

[11] M. Tenorth and M. Beetz. KnowRob — Knowledge Processing for Autonomous Personal Robots. In *IEEE/RSJ International Conference on Intelligent RObots and Systems.*, 2009.

[12] M. Tenorth, D. Nyga, and M. Beetz. Understanding and executing instructions for everyday manipulation tasks from the world wide web. In *IEEE International Conference on Robotics and Automation (ICRA). Accepted for publication.*, 2010.

[13] R. Urtasun and T. Darrell. Discriminative Gaussian process latent variable model for classification. In *Proceedings of the 24th international conference on Machine learning*, pages 927–934. ACM New York, NY, USA, 2007.

[14] J. Wang, D. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models. *Advances in Neural Information Processing Systems*, 18:1441–1448, 2006.