8

Manipulative Action Recognition for Human-Robot Interaction

Zhe Li, Sven Wachsmuth, Jannik Fritsch[1] and Gerhard Sagerer
Applied Computer Science, Faculty of Technology, Bielefeld University
Germany

1. Introduction

Recently, human-robot interaction is receiving more and more interest in the robotics as well as in the computer vision research community. From the robotics perspective, robots that cooperate with humans are an interesting application field that is expected to have a high future market potential. A couple of global and also mid-sized companies have come up with quite sophisticated robotic platforms that are designed for human-robot interaction. The ultimate goal is to place some robotic assistant or companion in the regular home environment of people, who would be able to communicate with the robot in a human-like fashion. As a consequence, the "hearing" as well as the "seeing" -- as the most prominent and equally important modalities -- are becoming major research issues.

From the computer vision perspective, robot perception is more than an interesting application field. During the last decades, we can note a shift from solving isolated vision problems to modeling visual processing as an integral connected component in a cognitive system. This change in perspective pays tribute to important aspects of understanding dynamic visual scenes, such as attention, domain and task knowledge, spatio-temporal context as well as a functional view of object categorization.

The visual recognition of human actions is in the center of all these aspects and provides a bridge for a non-verbal as well as verbal communication between a human and the robot, which both are highly ambiguous. It enables the robot's anticipation of human actions leading to a pro-active robot behavior especially in passive, more observational situations. Furthermore, it draws attention to manipulated objects or places, embeds objects in functional as well as task contexts, and focuses on the spatio-temporal dynamics in the scene.

Recently, much work has been done in the area of gesture-based human-robot interaction (HRI) because of humans' intensive use of their hands. These approaches mostly deal with *symbolic, interactional*, or *referential gestures* that have a communicative meaning on their own (Nehaniv, 2005). In terms of Bobick's taxonomy of *movements, activities, and actions* (Bobick, 1998) they can be characterized as movements or, in more structured cases,

[1] J. Fritsch is now with the Honda Research Institute Europe GmbH in Offenbach, Germany.

Source: Vision Systems: Segmentation and Pattern Recognition, ISBN 987-3-902613-05-9,
Edited by: Goro Obinata and Ashish Dutta, pp.546, I-Tech, Vienna, Austria, June 2007

activities. In this regard, object manipulations[2] are more complex because the hand trajectory needs to be interpreted in relation to the manipulated object. Due to Bobick this kind of context characterizes *actions*.

In this chapter, we aim at the vision-based recognition of simple actions that are defined by a non-deterministic sequence of object manipulations. As a manipulative gesture, this serves an important communicative function in human-robot interaction. First, the manipulation of an object draws the attention of the communication partner on the objects that are relevant for a performed task. Secondly, it serves the goal of a more pro-active behavior of the robot in passive, more observational situations. As Nehaniv states: "If the robot can recognize **what** humans are doing and to a limited extent **why** they are doing it, the robot may act appropriately" (Nehaniv, 2005). For example, in Fukuda's work a cooking support robot is developed (Fukuda et al., 2005). It can recognize human manipulations of objects by sensing the movements of the markers on the objects and give recommendations by speech or gesture. Dropping these kinds of artificial constraints, the recognition problem is becoming notoriously difficult. Assuming that a hand is manipulating a spatially near object, it becomes hard to decide if the object is just passed by the hand or manipulated. Besides this segmentation ambiguity, there is a large spatio-temporal variability of how hand trajectories reach different object types and the appearance of a hand trajectory in a 2D image will also heavily vary according to the position of the object and the view-angle. Finally, the mutual occlusion between the hand and the object causes even more difficulties for object detection and tracking.

In the present approach we will focus on two problems in the recognition of manipulative actions: (i) the segmentation ambiguity and (ii) spatio-temporal variability of the hand trajectory. We propose a unified graphical model with a two-layered recognition structure. On the lower layer, the object-specific manipulative primitives are represented as Hidden Markov Models (HMM) which are coupled with task-specific Markovian models on the upper level. A top-down processing mechanism predicts which kinds of objects are relevant according to the currently recognized tasks. Thereby, a dynamic attention mechanism is realized that reduces the number of considered objects and simplifies the segmentation task of the hand trajectory. Furthermore, the manipulative primitives are spotted by a particle filter (PF) realized HMM matching process. Due to an explicit modeling of an action abortion and resampling step, this method is more promising than traditional HMM forward-backward (Rabiner, 1990) processing and also could achieve more flexible transitions between model states than condensation-based trajectory recognition (Black & Jepson, 1998). Afterwards, the results are fed back into the task level in order to predict the following primitives closing the bottom-up and top-down cycle.

[2] Nehaniv refers to them as *manipulative gestures* (Nehaniv 2005).

Fig. 1. The Bielefeld Robot Companion (BIRON)

In the following part of this chapter, we will firstly review some related work in the field of human action and activity recognition. Then, we will present our system architecture which takes the temporal as well as the spatial context into account. The recognition of human actions is realized in a tightly coupled loop of bottom-up and top-down processing. We start by describing the low-level image processing of the bottom-up part. Then, we discuss how the object-specific manipulative primitives are spotted under spatio-temporal variability. The modeling of the manipulative task lies on top. The other half of the loop combines the top-down task knowledge with the bottom-up processing scheme. The experiment section presents the results on a corpus of 8 persons performing 3 different tasks consisting of different sequences of primitive actions. Finally, the conclusion will give some discussion on the approach and the possible future work.

2. Recognition of Human Movements, Activities, and Actions

A robot that is autonomously moving and acting in a human environment needs to understand and predict human behavior to a certain degree. While small automatic vacuum cleaners will mainly deal with collision avoidance for safety issues, larger movable robots, like the Bielefeld Robot Companion (Haasch et al., 2004) in Fig. 1 which is based on a Pioneer peopleBot platform, need to respect human activities and situations beyond physical predictability leading to the recognition of human intentions. This starts by considering social spaces, detecting when a person does not want to be disturbed, and ends in solving cooperative tasks with a human partner. The same accounts for human-robot communication starting with the problem to detect if and when a person communicates with the robot (Lang et al. 2003), via the interpretation of a communicative gesture (Pavlovic et al. 1997) to the interpretation of the action context of an unspecific verbal statement (Wachsmuth & Sagerer, 2002; Ballard & Yu, 2003). The reason for the increasing complexity in the interpretation of human motion patterns is the underlying factor of human intentions. The meaning of very similar human motions heavily depends on different levels of human intention. In this regard, Fleischman and Roy (2005) argue that learning the meaning of verbs is much harder that learning nouns. They distinguish between two different kinds of

ambiguities. (1) The vertical ambiguity refers to a possible causal chain of intentions, e.g. in order to *get a cup*, I need to *find a cup, open the cupboard,* and *grab the handle*. Thus, the same action *'the hand moves to the handle of the cupboard'* could be named on different levels of intention. (2) The horizontal ambiguity resambles that the high level interpretation could be ambiguous. For example, the same action as before could be interpreted as *clean the cupboard* instead of *get a cup.*

The different levels of intention have a different scope of interpretation in time and space. The physical prediction can be managed on a subsymbolic level considering the current trajectory of the human movement. Modeling social spaces needs at least some kind of representation of the human's mental state, while the recognition of actions like the opening of a cupboard needs to consider the relation of a human pose with regard to environmental objects and the changes of the object states over time.

The concept of different interpretation scopes directly fits Bobick's categorization of motion recognition: movement, activity, and action (Bobick, 1998). While movements can be characterized by reoccurring trajectories with a dedicated symbolic meaning, the interpretation of activities needs the extension of the scope in time in order to infer a higher level of intention. It represents larger-scale events, which typically include interactions with the environment and causal relationships. Actions involve a state change of the environment extending the scope into space.

So far we did not focus on the kind of body movement performed by a human. A large amount of work is dedicated to whole body movements. An overview of several approaches is given by Gavrila (1999). Spatial as well as temporal contexts are considered by Intille & Bobick (2001) in terms of multiperson actions and Fleischman, Decamp, & Roy (2006) in terms of places in an living environment. However, these approaches are mainly based on top-down views from surveillance cameras. In the robotics field most work is dedicated to *gestures,* i.e. intentional hand/arm movements that are mainly used for human-computer or human-robot interaction. A taxonomy of these is given by Pavlovic, Sharma, & Huang (1997). They distiguish between manipulative and communicative gestures, on the one hand, and unintentional movements, on the other hand. Manipulative gestures are used to act on objects in the environment and, thereby, constitute actions, while communicative gestures are mainly characterized by a temporally structured activity. In the following, we will focus on manipulative gestures.

The recognition of manipulative gestures is one of the most complex tasks as the system needs to extract relational features between the human motion and the environmental objects in cases of a high degree of occlusion. Therefore, most related work on manipulative action recognition simplifies the setting to a certain degree. A common scenario that is well motivated from domestic environments assumes that all relevant actions are performed on a table top (e.g. preparing a meal, decorating a table, performing typical office work, watering flowers). Thus, we assume that a mobile robot moves to a place around the table where it is able to observe the sequence of actions in focus.

In order to recognize these, more sophisticated schemes are needed that explicitly model contextual factors defining actions. Jo used a Finite State Machine (FSM) for modeling possible state transitions in the manipulative gesture recognition (Jo et al., 1998). Bobick developed a PNF (past-now-future) constraint network to model the temporal structure of actions and subactions (Pinhanez & Bobick, 1998). These typically are pure semantic approaches, which have not used explicit motion models. In Chan's work, a simple feature

vector is used for modeling the interaction primitive, e.g. *approach*. The transition of the semantic primitives are modeled by HMMs (Chan et al., 2004). Because of the early symbolic abstraction of trajectory information, this method can only be applied in a restricted scenario. An approach that actually combines both types of information, sensory trajectory data and symbolic object data, in a structured framework is Moore's concept of objectspaces (Moore et al., 1999). Here a camera mounted on the ceiling observes a human interacting with different objects. Certain image processing steps are carried out to obtain image-based, object-based, and action-based evidences for objects and actions, which are integrated using Bayesian networks. Action primitives are recognized from hand trajectories using HMMs that are trained offline on different activities related to the known objects. Our approach uses a similar object represention scheme but goes beyond this work because the spotting of meaningful parts in longer hand trajectories is seriously considered and a combined top-down and bottom-up mechanism solves the object attention problem. Furthermore, the proposed model enables the system to infer high-level intentions in the manipulative gesture detected.

While these approaches center a context area around detected objects, hand-centered methods define context areas relative to a hand trajectory. Fritsch et al. (2004) put forward such an approach. In this case, the trajectory information is augmented in each time step by contextual objects that are searched on-line using the context area bound to the moving hand. A hierarchical structure is used to model the manipulative sequence by Li et al. (2005). In both works, the segmentation and spatio-temporal variability problems are coped with a particle filter. But the hand trajectory template, which is used as the primitive, lacks the capability of generalization. For representing all possible hand trajectories in manipulation, a huge number of templates are needed.

Another specific application is presented by Yu & Ballard (2002). They argue that the eyes guide the hand in almost every action or object manipulation. In their work, the eye motion is measured by a head-mounted eye tracker and used for the segmentation of hand trajectories and the detection of objects. HMMs are used for action recognition which is purely based on trajectory information. Then object and action information is integrated on a symbolic level using action scripts.

3. System Architecture

In contrast to purely trajectory-based techniques, the presented approach is called object-oriented w.r.t. two different aspects: it is object-centered in terms of trajectory features that are defined relative to an object, and it uses object-specific models for action primitives. In our definition, the manipulative action has two semantic layers. The bottom layer consists of the object-specific manipulative primitives. Each object has its own set of manipulative primitives because we argue that different object types serve different manipulative functions and even manipulations with the same functional meaning are performed differently on different objects. The top layer is used for representing the manipulative task, which are modeled by typical transitions between certain manipulative primitives. The system architecture is shown in Figure 2. The architecture realizes a combined bottom-up top-down processing loop that utilizes the task-level prediction of possible primitives in order to restrict the object types possibly detected as well as the action primitives possibly recognized. In the bottom-up path, according to the top-down prediction a processing thread is created for each detected object that consists of a trajectory segmentation, a feature

computation, and an HMM-based recognition step. Thus, all three steps are performed differently for each object in parallel and the hand trajectory information is passed to each object-centered processing thread. The parallel processing for the objects avoids the ambiguity of the trajectory context if there are many objects in the scene. In the following sections, we will show how the object-specific manipulative primitives are detected in each thread, are combined for task recognition and effect the top-down process.

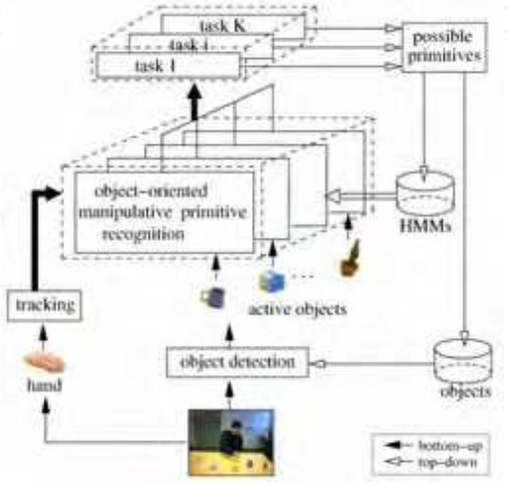

Fig. 2. The system architecture and the processing flow

4. Feature Extraction

The manipulative gesture is different to the face-to-face interactional gesture because the former reflects the interaction between the human's hand and the objects while the latter is typically characterized by a meaningful trajectory of the pure hand movement, e.g. the American Sign Language (Starner & Pentland 1995). Therefore, besides tracking the performing hand over time, the objects in the scene are also detected. For modeling typical object manipulations like "take" or "pour", the selected features describe the relative movements between the hand and the objects in 2D images. The reason why we are not using 3D representation is two fold. On the one hand, the 3D tracking of a person would need an elaborated body model and its tracking in mono-camera images is still a field of active research (Schmidt et al., 2006). Better tracking results can be achieved by using stereo cameras, which poses further constraints on the hardware setting. On the other hand, we argue that the perspective of a robot with regard to manipulative actions performed on a table top (as described in Section 2) can be assumed to be roughly stable, if the robot is able to chose an appropriate position relative to the human actor. In the following, this section will explain the computation for locating the hand and objects in the images and the construction of the interaction feature vector.

4.1 Hand Detection and Tracking

The hand is detected in a color image sequence by an adaptive skin-color segmentation algorithm (Fritsch, 2003) and tracked over time using Kalman filtering. Figure 3 shows the

screen shot of the processing from left to right: the raw image, the thresholded image indicating the skin-color pixels, and the region tracking. Currently only single hand manipulations are assumed. So the bigger skin-color region is labeled as face. The smaller is the hand. The hand observation O_t^{hand} is represented by the hand position $(h_x, h_y)_t$ at time t.

Fig. 3. The screen shots of hand tracking

4.2 Pre-knowledge and Detection of Object

Because the features of the manipulative gesture are based on the relative movements, a reliable detection of objects is crucial for the overall system performance. In order to avoid occlusion problems with interacting hands, we assume that a standard object recognizer, like those using Scale Invariant Feature Transform (SIFT) (Lowe, 2003), is applied on the static scene. Then, object-dependent primitive actions are always defined with regard to the object that is approached by the hand trajectory. If a moved object is applied to another object, the second object defines the object context. As we can have several static objects in the scene, the overall object observation vector contains multiple objects:

$$O^{obj} = \{O_1^{obj}, \cdots, O_i^{obj}, \cdots, O_L^{obj}\} \tag{1}$$

with

$$O_i^{obj} = (o_x, o_y, ID, o_h, o_w) \tag{2}$$

The observation vector of a detected object O_i^{obj} contains its position (o_x, o_y), a unique identifier ID for each different object type in the scene and its height o_h and width o_w.

4.3 Segmentation of Trajectory

It is common sense that the relative movement between hand and object contains less interaction features when they are far away from each other. A vicinity of an object is defined that is centered in the middle of the object detected. It is limited by the ratio β of its radius and the object size, which is shown by a blue circle in Figure 4. Based on this vicinity, a pre-segmentation step of the hand trajectory is performed that ignores irrelevant motions for primitive recognition. Considering the possible occlusions in manipulation and the uncertainty in moving an object, a segment is started when the hand enters the vicinity or when an object is detected and the hand is already in the vicinity (object put down into the

scene). It ends when the hand goes out of the object's vicinity or when the object is lost after the hand moves away (object has been taken). As a consequence, the trajectory is segmented differently based on the different objects in the scene. To handle this multi-observation problem, one processing thread is started for each detected object. In the following, the processing in a single thread will be introduced. There, the final segmentation is directly coupled with the recognition step.

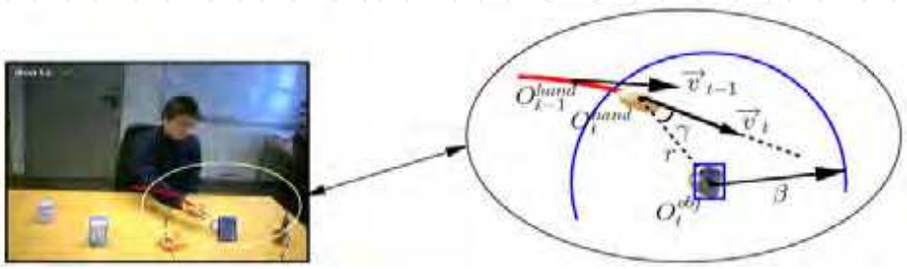

Fig. 4. The interaction feature vector

4.4 Interaction Feature Vector

During a manipulative action, the hand movements in the object vicinity can indicate an intended physical contact with object i, e.g. the hand will move towards the cup and slow down when the person wants to take it. Thus in the processing thread i, the interaction of the hand and the object is represented by a five-dimensional feature vector V_f that is calculated from O^{hand} and O_i^{obj}. It contains the features: magnitude of hand speed v, change of the hand speed Δv, change of speed direction $\Delta\alpha$, distance r between the object and the operative hand, as well as the angle γ of the line connecting object and hand relative to the direction of the hand motion.

$$V_f = (v, \Delta v, \Delta\alpha, r, \gamma) \tag{3}$$

The parameter v, Δv, and r are all scaled by object size. So the features are invariant with regard to translations, scale, and rotations.

5. Manipulative Primitive Detection

Although an object vicinity is defined for cutting away the hand trajectories which are less relevant to object manipulation, it is a coarse segmentation. The relative movements of the hand in an object vicinity can also contain both a typical interaction and some meaningless part. Consequently, the typical hand-object interactions, which we named *object-specific manipulative primitive*, have to be detected in a longer trajectory. The major methods include Dynamic Time Warping (DTW) (Alon et al., 2005), Artificial Neural Networks (ANN) (Kjeldsen & Kender 1995), and Hidden Markov Models (HMM) (Morguet & Lang, 1998; Lee & Kim, 1999). The DTW can to a certain extent cope with spatio-temporal variability. But as a template-based dynamic matching technique, it needs a large number of templates for a range of variations. ANN can achieve good detection results on static patterns,

including fixed length trajectories. It is not suited for the manipulative primitives which can have huge temporal variance. The HMM is another well-known technique for modeling sequential signals. By defining the transition between states and the state dependend observations in a probabilistic way, variations can be coped with to a certain degree. It is effectively used in speech recognition, handwriting recognition and human activities recognition. However, the standard forward algorithm to calculate the probabilities of the HMM candidates given the observation has the assumption that the whole sequence is emitted by one HMM. In order to spot the partition which conforms to an HMM from a long observation, some approaches, e.g. HMM-based threshold model (Lee & Kim, 1999) and Normalized Viterbi algorithm (Morguet & Lang, 1998) were put forward. Because the output score of the continuous observations of a given HMM will permanently increase or decrease, a window is used to tune the weights of the observation. Nonetheless, the fixed length of the window conflicts with the temporal variability of the signal. Recently the Sequential Monte Carlo (SMC) method also named Particle Filter (PF) is getting more and more focus in the pattern recognition society, which allows an on-line approximation of probability distributions using samples (particles). It has been used for template-based trajectory matching (Blake & Jepson, 1998). In order to keep the spatio-temporal variability of HMMs and use the advantage of PF on tracking the models with weighted particles, a PF realized HMM matching method is implemented to detect object-specific manipulative primitives. This process is building the bridge between the low-level image processing and the task knowledge.

5.1 HMM for Manipulative Primitive

The object-oriented manipulative primitives are modeled by ergodic HMMs. Different to the normal parameter set $\lambda = (A, B, \Pi)$ of an HMM, a terminal probability E is added. It reflects the terminal probability of an HMM given a hidden state s_i. So the whole set consists of:

- $\Pi = \{\pi_i | \pi_i = P(q_1 = s_i)\}$, initial probability of state s_i.
- $A = \{a_{ij} | a_{ij} = P(q_{t+1} = s_j | q_t = s_i)\}$, transition probability from state s_i to s_j.
- $B = \{b_i(k) | b_i(k) = P(o_t = v_k | q_t = s_i)\}$, probability of observing v_k given hidden state s_i.
- $E = \{e_i | e_i = P(q_{end} = s_i)\}$, terminal probability of state s_i.

Considering the small amount of training data, we use discrete HMMs. The whole feature space is discretized into 2x 2 x 4 x 3 =48 cells based on the following quantized dimensions:

Parameters	Quantization
v	$< v_{threshold}, \geq v_{threshold}$
Δv	$< 0, \geq 0$
$\Delta\alpha$	$< 90, \geq 90$
r	$[0 \ldots \beta/4 \ldots \beta/2 \ldots 3\beta/4 \ldots \beta]$
γ	$< 90, \geq 90$ if $v \geq v_{threshold}$

Table 1 Vector quantization of the interaction feature space

They define the observation states for the following HMMs. The angle γ between the object -hand connection line and the direction of the hand motion is quantizied conditioned on v because it has much noise when the hand speed is very low. The HMM parameter set is learned from manually segmented trajectories with the Baum-Welch algorithm, e_i is calculated similar to π_i, except using the last states.

5.2 PF-based HMM Matching

In order to detect the primitives from the pre-segmented trajectories, a PF called Sampling Importance Resampling (SIR) is used, better known as Condensation introduced by Isard and Blake (Isard & Black 1996). Figure 2 shows a two time-slice Dynamic Bayesian Network (DBN) which indicates the dependency structure of the probabilistic model. For each one in the L objects, the matching of the M HMMs and the observation are achieved by temporal propagation of a set of weighted particles:

$$\{(S_t^{(1)}, w_t^{(1)}), \cdots (S_t^{(N)}, w_t^{(N)})\} \tag{4}$$

with

$$S_t^{(i)} = \{p_t^{0(i)}, q_t^{(i)}, e_t^{(i)}\} \tag{5}$$

The number of particles is N. The sample $S_t^{(i)}$ contains the primitive index $p_t^{0(i)}$, the hidden state $q_t^{(i)}$, and the terminal state of this primitive $e_t^{(i)}$ at time t (see Figure 5). The resampling step reallocates a certain fraction of the particles with regard to the initial distribution Π. Consequently, the weight $w_t^{(i)}$ of a sample can be calculated from

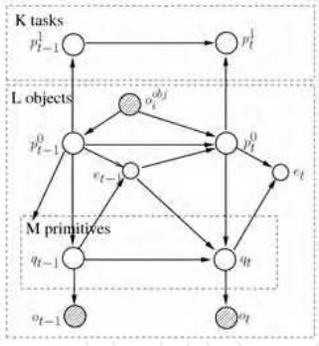

Fig. 5. A Dynamic Bayesian Network represents the dependency structure of two time slices in the recognition model. Each object-centered processing thread corresponds to one of the L plates in the dependency model. K is the number of different tasks modeled in the system and M is the number of possible primitives which each corresponds to one state of the variables p_t^0 and p_t^1, respectively. The upper index of these variables denotes the primitive vs. task level.

$$w_t^{(i)} = \frac{p(o_t \left| S_t^{(i)} \right.)}{\sum_{j=1}^{N} p(o_t \left| S_t^{(j)} \right.)} \tag{6}$$

The $p(o_t \left| S_t^{(i)} \right.)$ in it is the observation probability of o_t given $q_t^{(i)}$ and HMM $p_t^{0(i)}$. The propagation of the weighted samples over time consists of three steps:

Select: Selection of $N - M$ samples $S_{t-1}^{(i)}$ according to their respective weight $w_{t-1}^{(i)}$ and random initialization of M new samples. That means some particles which have high weights will be selected multiple times and some particles which have low weights will not be selected at all.

Predict: The current state of each sample $S_t^{(i)}$ is predicted from the samples of the select step according to the graphical model given in Figure 5. The terminal state $e_{t-1}^{(i)}$ is a bi-valued variable, 0 means the primitive is continuing and 1 means the primitive ends here. So if $e_{t-1}^{(i)}$ is 0, the next hidden state $q_t^{(i)}$ is sampled according to the transition probability of the HMM of primitive $q_{t-1}^{(i)}$ and the primitive index $p_t^{0(i)}$ keeps the same as $p_{t-1}^{0(i)}$. If the terminal state $e_{t-1}^{(i)}$ is 1, the primitive index $p_t^{0(i)}$ will be sampled according to the current possible primitives of this object. Then the hidden state $q_t^{(i)}$ is sampled according to the initial probability of the HMM of the new primitive $p_t^{0(i)}$. At the end of this step, the terminal state of this particle $e_t^{(i)}$ is sampled based on the terminal probability of the current primitive state $q_t^{(i)}$.

Update: Determination of the weights $w_t^{(i)}$ of the predicted samples $S_t^{(i)}$ using Eq. 6.

The recognition of a manipulative primitive is achieved by calculating the **end-probability** P_{end} that a certain HMM model p_i is completed at time t:

$$P_{end,t}(p_i) = \sum_n w_t^{(n)} \text{ , if } p_i \in S_t^{(n)} \tag{7}$$

A primitive model is considered recognized if the probability $P_{end,t}(p_k)$ of the primitive model p_k exceeds a threshold p_{th}^0 which has been determined empirically.
The resampling step in the particle propagation is able to adapt the starting point of the model matching process if the beginning of the primitive does not match the beginning of the segment. The end-probability gives an estimation of the primitive's ending point. This combination to a certain extent solves the problem of the forward-backward algorithm which needs a clear segmentation of the pattern.

6. Task Level Processing

6.1 Model of Tasks

The manipulative tasks are modeled as the first-level Markovian process which is the same as Moore's definition (Moore et al., 1999). Although this assumption violates certain domain dependencies, it is an efficient and practical way to deal with task knowledge. In the model Λ_i for a manipulative task i, a set of possible manipulative primitives P_i^1 are defined, e.g., in the "prepare tea" task, the primitives "take cup", "take tea can" could appear but not "take milk". Because of the high effort needed for recording a huge mount of task sequences, the number of training examples for each complete task is too low for robustly estimating transition probabilities. Therefore, we model a task by a set of possible primitive pair transitions similar to a word pair grammar in automatic speech recognition. The set of transition rules A_i^1, the possible start symbols Π_i^1, and the set of possible end symbols E_i^1 is learned from the output of the primitive recognition layer on a training set by thresholding the frequency of pairs observed in sequences of action primitives (see Section 7.2 for more details). Suppose the result from the manipulative primitive recognition is the sequence $p_1^1,\ldots,p_t^1$. To calculate the acceptance of a task $\Lambda_i = (P_i^1, \Pi_i^1, A_i^1, E_i^1)$, only the primitives which are in the primitive list of the task Λ_i will be chosen because of the possible insertion in the primitive recognition.

$$(p_1^1,\ldots p_t^1 \mid p_j^1 \in P_i^1, j=1\ldots t) \in \{\mathrm{P} \mid p_1^1 \rightarrow^*_{A_i^1} p_t^1, p_1^1 \in \Pi_i^1, p_t^1 \in E_i^1\} \qquad (8)$$

where P denotes the possible sequences from primitive p_1^1 to p_t^1 while considering transitions in A_i^1. Eq. 8 can easily be evaluated according to the parameter set Λ_i.

6.2 Top-down Process

Because of the object-specific primitive definition and its parallel processing for each affected object, the system confronts an attention problem when there are many objects appearing in the scene, simultaneously. In order to solve this problem, a top-down process is introduced, in which the possible subsequent primitives are predicted on the ground of the active task models and the previous results from the manipulative primitive recognition. This prediction is similar to the computation of a look ahead symbol in parsing strategies. For the prediction step, different parsing alternatives are considered during the HMM matching process. For all primitives that gain an end probability $P_{end,t}(p_i) > 0$ a lookahead symbol is generated. If a primitive has been recognized this primitive is eliminated as a lookahead symbol. Because the predicted action primitives are specific for certain object types, the set of the next possibly manipulated object types can be calculated and be passed to the object detection component. This realizes a task driven attentional cue for early processing steps of the system (Figure 2). Additionally, the expectations from the predicted action primitives are used to restrict the HMMs applied in the PF based matching process.

7. Experiments and Results

In order to evaluate the quality of the manipulative gesture recognition, a scenario in an office environment has been designed as shown in Figure 6. A person is sitting behind a table and manipulates the objects that are located on it. She or he is assumed to perform one of three different manipulation tasks:
(1) *water plant*: take cup, water plant, put cup;
(2) *prepare tea*: consists of take/put cup, take tea can, pour tea into cup, put tea can;
(3) *prepare coffee*: consists of take/put cup, take milk/sugar, pour milk/take sugar into cup, put milk.
In the experiment, each task is performed 4-5 times by 8 different persons resulting in 36 sequences for each task and a total of 108 sequences. The images are recorded with a resolution of 320x240 pixels and with a frame-rate of 15 images per second. The object recognition results have been labeled because the evaluation experiment should concentrate on the performance of the action and task recognition. The object in the hand is ignored so that *pour milk into cup* and *pour tea into cup* are the same primitive actions. The scenario is restricted in so far that we assume a static camera, a known configuration of objects, and a camera view that is roughly orthogonal to the relevant movements.

Fig. 6. The office scenario used in the experiment.

7.1 Manipulative Primitive Recognition

The first evaluation is used to test the performance of the object-oriented manipulative primitive recognition. There are five different objects used in the experiment: tea can, milk, sugar, cup and plant. Figure 7 shows the primitives defined for each object type. The evaluation is done for all segments computed by the pre-segmentation step (see Section 4.3). These segments either contain a real manipulative primitive action which we call positive segments (PS) or contain just a hand passing by an object which we call negative segments (NS). For the positive segments, we calculate the false negative (FN) rate. For negative segments, the false positive (FP) rate is calculated. In order to achieve a good system performance both rates should be low because both kinds of errors would seriously affect human-robot interaction. We randomly divided the 108 whole task sequences into a training set of 60, and a test set of 48 sequences. Because of the low number of training examples, we

run the Baum-Welch algorithm used for the HMM learning procedure 10 times with random initialization and give a standard deviation for the FN and FP rates. The results are computed using the parameter setting: $N = 500$, $M = 50$, $p_{th}^{0} = 0.2$, and $\beta = 3$. From the results shown in Figure 7, it could be found that the "put" primitives are recognized with lower FN rate than the "take" and "pour" primitives because the variations of the latter two are much higher from person to person. Figure 8 shows the end probabilities of different manipulative primitives in a prepare tea task. The horizontal line above zero is the recognition threshold and the temporal periods which are coloured indicate that the hand is in the object vicinity at that moment.

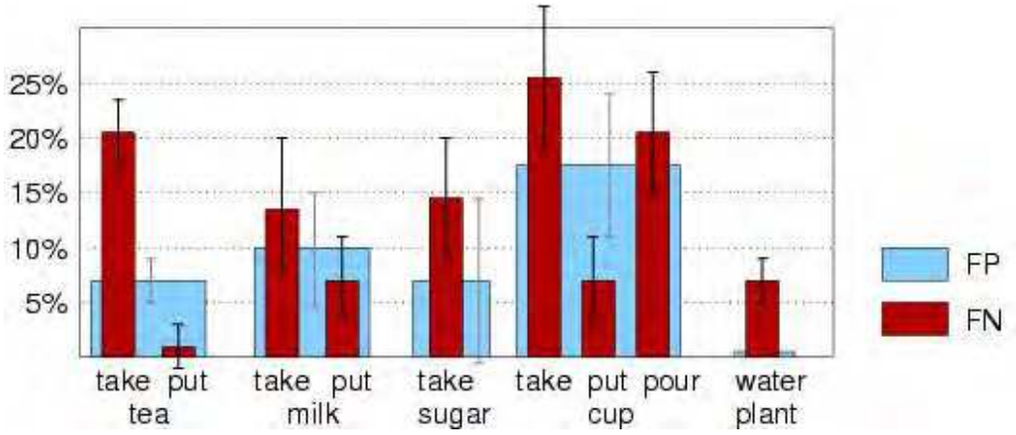

Fig. 7. The recognition results of the object-specific manipulative primitives in both positive and negative segments.

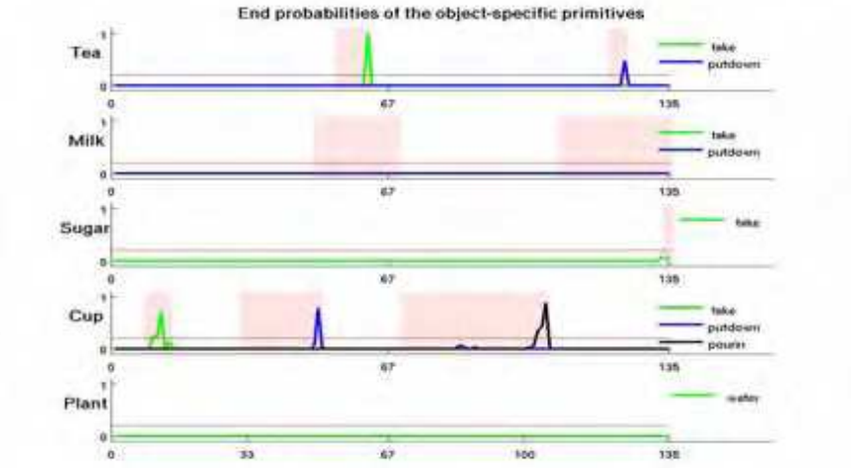

Fig. 8. The end probabilities of the object-specific manipulative primitives in a *prepare tea* task

7.2 Manipulative Task Recognition

The second evaluation assesses the overall system performance. A manipulative task consists of the manipulative primitive sequence. However the ordering of the sequence is neither pre-determined nor completely fixed. For example some people may take sugar before taking milk, some will do it the other way around. But there probably will be an ordering between taking the cup and the watering action which needs to be learned from the data. For learning the possible transition pairs of each task model, the data set is divided into the set of 20 observation sequences, that was already used for learning the primitive action models, and a set of 16 sequences that are used for a one-leave-out experiment. Thus, each task model is learned from 35 task sequences in each experiment. The possible word pair transitions are extracted from the training data by a frequency threshold. The task

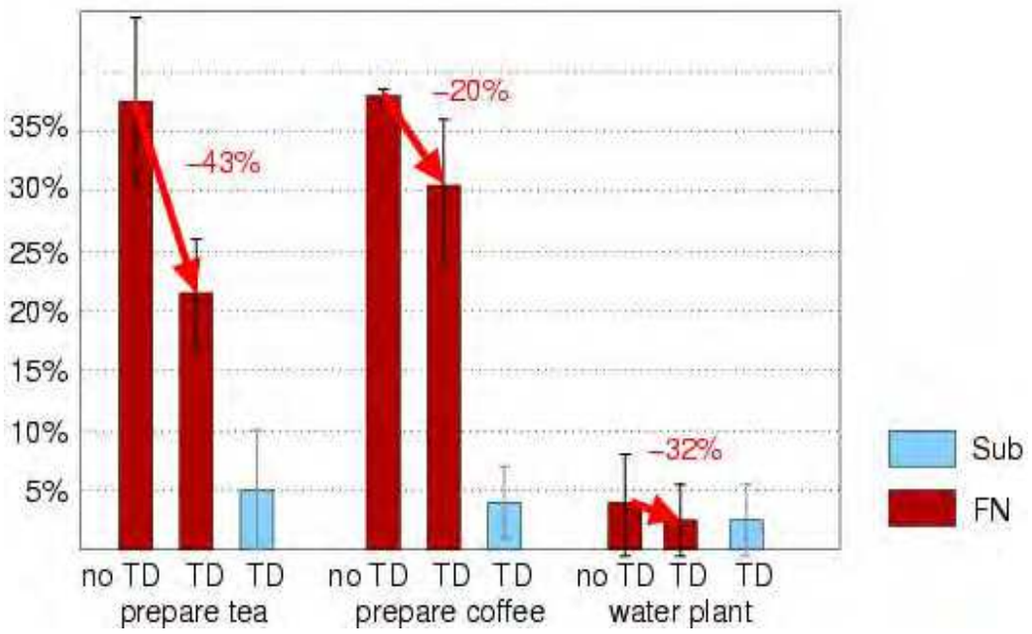

Fig. 9. The recognition results of the manipulative tasks with and with out top-down processing.

recognition results of the whole system are compared with (TD) and without (no TD) the top-down attention processing (see Figure 9). The FN rate clearly shows a significant drop in case of top down processing for *prepare tea* and *prepare coffee*. Because sometimes an expected primitive was misrecognized in a way that was not covered by the task grammar, the rejection of these tasks caused relatively high FN rates but nearly no substitution errors (Sub.). The processing time for a 180-frame ``prepare coffee'' sequence with the former method is 54s running on MATLAB, which is much lower than the 86s needed by the pure bottom-up processing.

8. Conclusion

The recognition of manipulative actions and tasks is an essential component for the natural, pro-active, and non-intrusive interaction between humans and robots. However, most techniques for the recognition of symbolic, interactional or referential gestures cannot be transferred because they ignore the object context and assume an object independent characteristic of the hand trajectory. Other approaches that focus on action recognition either use a pure semantic approach without considering motion models or simplify the trajectory segmentation problem in a pure bottom-up process.

The presented approach overcomes several of these deficiencies. The contextual objects are used for a pre-segmentation of the hand trajectory; the manipulative action primitives are spotted by a particle filter approach that matches object specific HMMs in a more flexible way than the traditional forward-backward algorithm; tasks are defined by a set of possible transition rules similar to a word pair grammar that is automatically extracted from a small test set. By calculating a set of lookahead symbols on the task level, a task-driven attention filter is realized that tightly couples bottom-up and top-down processing. We were able to show first experiments that underline the potential of the presented approach. The action primitives were recognized quite robustly. The top-down attention filter significantly improves the computation time as well as the recognition performance.

Further work needs to concentrate on several issues. In terms of feature description neither pure symbolic nor trajectory-based characterizations will be general enough to describe the huge variety of manipulative actions. Trajectory-based features allow to distinguish actions that do not result in observable state changes of the objects, but suffer from large trajectory variations. The proposed object specific motion-models account to these variations to a certain degree. How to deal with multiple representations on both symbolic and sub-symbolic levels is still an open research question. The coupling of motion models and object types also leads to another important aspect of actions: the concept of object affordances. The observed shape and function of an object activates an expected set of hand trajectories and vice versa. We expect that this kind of coupling will be a key issue both in categorization of objects and learning new action verbs. Another aspect is the development of more sophisticated task models that need to include human intentions on multiple scopes of time and space. Finally, more sophisticated experiments are needed to evaluate current action recognition approaches. Appropriate benchmark datasets for manipulative action recognition are currently not available and most approaches focus on their specific application domain.

9. References

Alon, J.; Athitsos, V. & Sclaroff, S. (2005) Accurate and efficient gesture spotting via pruning and subgesture reasoning. In *Proc. ICCV Human-Computer Interaction Workshop*, pages 189–198.

Black, M. J. & Jepson, A. D. (1998). A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *European Conf. On Computer Vision, ECCV-98*, pages 909–924, Freiburg, Germany.

Ballard, D. H. & Yu, C. (2003). A multimodal learning interface for word acquisition. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing.* volume 5, pages 784–790.

Bobick, A. F. (1998). Movement, activity, and action: The role of knowledge in the perception of motion. In *Royal Society Workshop on Knowledge-based Vision in Man and Machine*.

Chan, M.T.; Hoogs, A.; Schmiederer, J. & Petersen, M. (2004). Detecting rare events in video using semantic primitives with hmm. In *ICPR04*, pages IV: 150–154.

Fleischman, M.; Decamp, P. & Roy, D. (2006). Mining temporal patterns of movement for video content classification. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 183–192, New York, NY, USA. ACM Press.

Fleischman, M. & Roy, D. (2005). Why verbs are harder to learn than nouns: Initial insights from a computational model of intention recognition in situated word learning. In *Proc. of the 27th Annual Meeting of the Cognitive Science Society*.

Fritsch, J. (2003). *Vision-based Recognition of Gestures with Context*. Dissertation, Bielefeld University, Technical Faculty.

Fritsch, J.; Hofemann, N. & Sagerer, G. (2004). Combining Sensory and Symbolic Data for Manipulative Gesture Recognition. In *Proc. IEEE ICPR*, pages 930–933, Cambridge, UK.

Fukuda, T.; Nakauchi, Y.; Noguchi, K. & Matsubara, T. (2005). Time series action support by mobile robot in intelligent environment. In *Proc. IEEE Int'l Conf. Robotics and Automation*, pages 2908–2913, Barcelona, Spain.

Gavrila, D. M. (1999). The visual analysis of human movement: a survey. In *Comput. Vis. Image Underst.*, 73(1):82–98.

Haasch, A.; Hohenner, S.; Huwel, S.; Kleinehagenbrock, M.; Lang, S.; Toptsis, I.; Fink, G. A.; Fritsch, J.; Wrede, B.; & Sagerer, G. (2004). Biron — the bielefeld robot companion. In *Proc. Int. Workshop on Advances in Service Robotics*, pages 27–32, Stuttgart, Germany.

Intille, S. S. & Bobick, A. F. (2001). Recognizing planned multiperson action. In *Comput. Vis. Image Underst.*, 81(3):414–445, March 2001.

Isard, M.; & Blake, A. (1998). Condensation – conditional density propagation for visual tracking. In *Int. J. Computer Vision*, pages 5–28.

Jo, K. H.; Kuno, Y. & Shirai, Y. (1998). Manipulative hand gesture recognition using task knowledge for human computer interaction. In *Proc. Int'l Conf. on Automatic Face and Gesture Recognition*, pages 468–473.

Kjeldsen, R. & Kender, J.R. (1995) Visual hand gesture recognition for window system control. In *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, pages 184–188.

Lang, S.; Kleinehagenbrock, M.; Hohenner, S.; Fritsch, J.; Fink, G. A. & Sagerer, G. (2003). Providing the basis for human-robot-interaction: a multi-modal attention system for a mobile robot. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 28–35, New York, NY, USA, ACM Press.

Lee, H. K. & Kim, J. H. (1999). An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973.

Li, Z.; Hofemann, N.; Fritsch, J. & Sagerer, G. (2005). Hierarchical modeling and recognition of manipulative gesture. In *Proc. ICCV, Workshop on Modeling People and Human Interaction*, Beijing, China.

Li, Z.; Fritsch, J.; Wachsmuth, S. & Sagerer, G. (2006). An object-oriented approach using a topdown and bottom-up process for manipulative action recognition. In *Annual Symposium of the German Association for Pattern Recognition (DAGM),* pages 212–221, Berlin, Germany, Springer-Verlag.

Lowe, D. (2003). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 20:91–110.

Moore, D.J.; Essa, I.A. & Hayes III, M.H. (1999). Exploiting human actions and object context for recognition tasks. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 20–27.

Morguet, P. & Lang, M. (1998). Spotting dynamic hand gestures in video image sequences using hidden markov models. In *International Conference on Image Processing*, pages 193–197, Chicago, USA.

Nehaniv, C. P. (2005). Classifying types of gesture and inferring intent. In *Proceedings of the Symposium on Robot Companions: Hard problems and Open Challenges in Robot-Human Interaction AISB'05*, pages 74–81, Hatfield, UK.

Pavlovic, V.; Sharma, R. & Huang, T. S. (1997). Visual interpretation of hand gestures for humancomputer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695.

Pinhanez, C. S. & Bobick, A. F. (1998). Human action detection using pnf propagation of temporal constraints. In *Proc. IEEE CVPR*, pages 898–907, Washington, DC, USA.

Rabiner, L. R. (1990). A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Schmidt, J.; Kwolek, B. & Fritsch J. (2006). Kernel particle filter for real-time 3D body tracking in monocular color images. In *Proc. of Automatic Face and Gesture Recognition*. Pages 567-572, Southampton, UK.

Starner, T. & Pentland, A. (1995). Real-time american sign language recognition from video using hidden markov models. In *IEEE International Symposium on Computer Vision*, pages 265–270.

Yu, C. & Ballard, D. H. (2002). Learning to Recognize Human Action Sequences. In *2nd International Conference on Development and Learning (ICDL'02)*, pages 28–34.

Wachsmuth, S. & Sagerer, G. (2002) Bayesian networks for speech and image integration. *In Eighteenth national conference on Artificial intelligence*, pages 300–306, Menlo Park, CA, USA.

Vision Systems: Segmentation and Pattern Recognition
Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-05-9
Hard cover, 536 pages
Publisher I-Tech Education and Publishing
Published online 01, June, 2007
Published in print edition June, 2007

Research in computer vision has exponentially increased in the last two decades due to the availability of cheap cameras and fast processors. This increase has also been accompanied by a blurring of the boundaries between the different applications of vision, making it truly interdisciplinary. In this book we have attempted to put together state-of-the-art research and developments in segmentation and pattern recognition. The first nine chapters on segmentation deal with advanced algorithms and models, and various applications of segmentation in robot path planning, human face tracking, etc. The later chapters are devoted to pattern recognition and covers diverse topics ranging from biological image analysis, remote sensing, text recognition, advanced filter design for data analysis, etc.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen