paco plus

perception, action and cognition
through learning of object-action complexes

Project no.:	**IST-FP6-IP-027657**
Project full title:	**Perception, Action & Cognition through Learning of Object-Action Complexes**
Project Acronym:	**PACO-PLUS**
Deliverable no.:	**D3.2.3**
Title of the deliverable:	**Aspects of integrating hand and upper body tracking for grasp activity recognition**

Contractual Date of Delivery to the CEC:	31 January 2009	
Actual Date of Delivery to the CEC:	31 January 2009	
Organisation name of lead contractor for this deliverable: Royal Institute of Technology (KTH)		
Author(s): Hedvig Kjellström, Danica Kragic, Volker Krüger		
Participants(s): KTH, AAU		
Work package contributing to the deliverable:	WP3.2	
Nature:	R	
Version:	1.0	
Total number of pages:	70	
Start date of project:	1st Feb. 2006	**Duration:** 48 month

Projectco-funded by the European Commission within the Sixth Framework Programme (2002-2006)
Dissemination Level

PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This report contains several scientific publications that are relevant for deliverable D3.2.3. Deliverable D3.2.3 is concerned with deriving parameters from video, describing the pose of a human, for the purpose of visual recognition and and human-to-robot mapping of manipulation and grasping activity. The included publications discuss different aspects of this area, and their relationship is outlined in an enclosed summary.

Keyword list: Action Recognition, Grasp Recognition, Human-to-Robot Mapping, Computer Vision, Robotics, AI

Summary

This deliverable consists of six conference papers [1, 2, 3, 4, 5, 7] and one technical report [6], describing a student project performed during the spring of 2008. Early versions of the papers [3, 4] were referred to in the PACO-PLUS activity report Feb 2007 – Jan 2008, and have since then been published.

The objective of WP3.2 is to define a representation of human hand activity so that a robot can observe the grasping activity being performed, map this activity to its own embodiment and perform the corresponding activity with its own manipulator(s). The application of this in the PACO-PLUS project is visual robot learning from demonstration.

1 Recognition-Based Approach to Body and Hand Pose Extraction

The subject of this report is tracking of humans in video for the purpose of recognizing grasp and manipulation activities and mapping them to a robot. Recognition of these types of activities is to a great extent based on the human's hand motion. However, to be robust, articulated[1] hand tracking should rely on estimation of the overall human pose, since the pose of the body constrains the estimation of hand pose. It is therefore desirable to track the full human (upper) body for grasp and manipulation activity recognition purposes.

Articulated tracking of the human body from monocular or small-baseline stereo video is a highly underconstrained and non-trivial problem. Additional complications are added if human body parts of different spatial scales, such as fingers and torso, are to be tracked simultaneously; normal digital-video resolution is not high enough to successfully capture small (e.g., finger) movements while keeping the whole human in view.[2]

In WP3, we have therefore approached the articulated tracking problem in a recognition-based manner [1, 2, 3]. The key idea is that human motion is highly constrained; the actual number of degrees of freedom of human motion is much lower than the dimensionality of the space spanned by all joint angles in the body. Two things can be observed. Firstly, humans do not exploit all physically possible joint configurations, they instead move according to quite specific patterns, often displaying symmetries, as in walking for example. Secondly, human motion is to a large extent defined by the type of activity the human is involved in; this is termed "tracking in action space" in WP3.1.3, WP3.2.3.

The idea behind the recognition-based upper-body tracking [1, 2] is to classify the type of action taking place (e.g., pointing), and then parameterize it (e.g., tune the pointing direction) so that the human action model fits with the human motion visible in the video sequence. In this way, the human is tracked, although only within limits permitted by the action in which the human is involved. A Parametric Hidden Markov Model (PHMM) [2] is used, and the method is evaluated on reaching and pointing actions, where the parameterization steers the hand motion direction or the hand location during the course of the action.

A method for classification of hand shapes into grasps has also been developed [3]. This method

[1] Here, articulated tracking means estimation of the motion of individual limbs.

[2] This problem is addressed in WP2.1 using attention and scene exploration with fovated vision.

can be seen as a crude estimator of articulated hand pose, as each grasp type represents a certain pose.[3] A grasping hand is represented as a Histogram of Oriented Gradients (HOG), a image representation of shape, capturing finger configuration and orientation to a certain extent. A new hand image is classified as one of six grasps by an approximate k-Nearest Neighbor (kNN) search among large set of synthetically generated hand images. For approximate kNN search, we use Locality Sensitive Hashing (LSH). The dataset contains grasps from all expected viewpoints and with expected occlusion. This makes the method view-independent although no 3D representation of the hand is computed.

The integration of these methods into a system for tracking human body parts on multiple scales is very straight-forward: The recognition-based upper-body tracking [1, 2] gives global hand positions in each frame. These estimated global hand positions are then used as a starting point for the representation of local hand shape [3].

If the goal is to track the hand only, an alternative to tracking the upper body is to use a segmentation approach, where each pixel in the image is assigned to either hand, rest-of-body, manipulated-object, or background. In this way, the hand position is more robustly estimated than with, e.g., skin segmentation only. Initial investigations of this approach [6] have shown promise, and we will continue to explore it further.

2 Human-to-Robot Grasp Mapping

The recognition-based hand pose estimator is incorporated in a human-to-robot grasp mapping system, in which a robot observes the human grasp, classifies it, and selects between a number of predefined grasping strategies [3, 7]. Each robot grasp strategy is parameterized by the position and orientation between the hand and the object, detected along with the grasp.

Thus, the 3D tracking and reconstruction of the human hand is implicit. Instead of reconstructing the human hand and mapping the reconstruction to the robot hand, a ”partial reconstruction”, the type of grasp, is obtained. The grasp class is then used to reconstruct the full pose of the robot hand. The mapping is made in grasp space instead of the 3D pose space. This corresponds to the human pose reconstruction in ”action space” discussed above [1, 2].

3 Manipulation Action Recognition

The HOG representation of hand shape discussed above is also employed in temporal segmentation and classification of human manipulation actions [4, 5].

The mapping from human to robot motion is in Section 2 restricted to a motor level. If the robot is humanoid, i.e., has an embodiment very similar to a human, this is adequate for learning to perform a task consisting on a sequence of hand actions. However, in the general case one cannot assume the robot to have a human-like embodiment. For example, the number of degrees of freedom in the robot's hands and arms, or even the number of robot arms or fingers, might differ from a human. In this case, the human motion has to be interpreted on a more abstract task level, rather than on a motor evel, in order to make a mapping to the robot possible.

There is also a statistical correlation between types of objects, object shapes, human hand actions, and human grasps in a Learning from Demonstration scenario, in analog to the OAC concept. We therefore recognize human manipulating actions and manipulated objects in parallel, letting the classifications of objects and actions influence each other within a Conditional Random Field. Human hand actions are represented as sequences of 2D hand poses and HOGs, as in Section 2. Objects are represented by sequences of HOGs. Results show the benefit of modeling objects and actions together, as opposed to recognizing them independently.

[3] The extreme of this is to have a continuous hand pose space, and to do regression into this space.

Attached Papers

[1] D. Herzog, A. Ude, V. Krüger, Motion imitation and recognition using parametric hidden markov models, in *IEEE International Conference on Humanoid Robots*, 2008.

[2] D. Herzog, V. Krüger, D. Grest, Parametric hidden Markov models for recognition and synthesis of movements, in *British Machine Vision Conference*, pp 163–172, 2008.

[3] H. Kjellström, J. Romero, D. Kragic, Visual recognition of grasps for human-to-robot mapping, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 3192–3199, 2008.

[4] H. Kjellström, J. Romero, D. Martínez, D. Kragic, Simultaneous visual recognition of manipulation actions and manipulated objects, in *European Conference on Computer Vision*, vol 2, pp 336–349, 2008.

[5] H. Kjellström, D. Kragic, *Visual Object-Action Recognition for Learning from Demonstration*, in submission 2009.

[6] D. Martínez, J. Romero, H. Kjellström, D. Kragic, *Image Segmentation of Manipulation Actions with Level Sets*, Technical Report TRITA-CV 2008:2, KTH 2008.

[7] J. Romero, H. Kjellström, D. Kragic, *Markerless Human-to-Robot Grasp Mapping based on a Single View*, in submission 2009.

Motion Imitation and Recognition using Parametric Hidden Markov Models

Dennis Herzog [1], Aleš Ude [2,3], Volker Krüger [1]

[1]*Aalborg University Copenhagen, Denmark* deh@cvmi.aau.dk, vok@cvmi.aau.dk

[2]*Jožef Stefan Institute, Slovenia* ales.ude@ijs.si

[3]*ATR Computational Neuroscience Laboratories, Japan* aude@atr.jp

***Abstract*—The recognition and synthesis of parametric movements play an important role in human-robot interaction. To understand the whole purpose of an arm movement of a human agent, both its recognition (e. g., pointing or reaching) as well as its parameterization (i. e., where the agent is pointing at) are important. Only together they convey the whole meaning of an action. Similarly, to imitate a movement, the robot needs to select the proper action and parameterize it, e. g., by the relative position of the object that needs to be grasped.**

We propose to utilize parametric hidden Markov models (PHMMs), which extend the classical HMMs by introducing a joint parameterization of the observation densities, to simultaneously solve the problems of action recognition, parameterization of the observed actions, and action synthesis. The proposed approach was fully implemented on a humanoid robot HOAP-3. To evaluate the approach, we focused on reaching and pointing actions. Even though the movements are very similar in appearance, our approach is able to distinguish the two movement types and discover the parameterization, and is thus enabling both, action recognition and action synthesis. Through parameterization we ensure that the synthesized movements can be applied to different configurations of the external world and are thus suitable for actions that involve the manipulation of objects.

I. Introduction

To effectively interact with people, a humanoid robot needs to be able, firstly, to recognize human movements in order to understand the intentions of an agent communicating with the robot, and secondly, to imitate human actions in a human way. These are essential tools that are needed to enable the robot to autonomously operate in a human environment.

We are interested in an action representation that is (a) easily and efficiently trained by demonstrating a set of exemplar movements and (b) able to recognize and synthesize the demonstrated actions for arbitrary parameterizations. The action representation should allow the robot to recognize and perform the learned movements also for not previously demonstrated parameterizations, e. g., reaching and pointing at objects at arbitrary locations.

To achieve this goal we utilize parametric hidden Markov models (PHMMs), which were originally proposed in [1]. We encode PHMMs by observations states that consist of Cartesian 3-D positions of shoulder, sternum, elbow, wrist, thumb, index-finger and its knuckle. The advantage of such an representation is that the goal of the considered actions (reaching and pointing) is explicitly encoded in the final state of the PHMMs, which makes it easier to find parameterization for action interpolation. To assure a proper interpolation that preserves the shape of the action trajectory, a proper alignment of the different exemplar actions, i. e. the alignment of corresponding hidden Markov states, is essential. We solved this problem by constraining the state transitions.

As an example, we studied the teaching of a humanoid robot through pointing and reaching gestures. The goal was to put differently shaped objects into a children's toy box with differently shaped holes corresponding to different shapes. The robot should learn both symbolic knowledge (which object belongs to which hole) and continuous action knowledge (how to move the objects and release them into the appropriate hole). For training and testing, the objects could be placed at any location on the table. An online demo is available via web page [2].

In addition, we present a systematic evaluation of the recognition and synthesis of our action representation. In the evaluation, we focus on the actions used in the demonstration on a humanoid robot HOAP-3, i.e. reaching out for an object to grasp it and pointing actions. Both actions have very similar trajectories (starting and ending in the same base pose) and are thus difficult to distinguish. It is interesting to note that simple diagnostic features like arm velocity, or the distance from hand to chest would fail in this context.

In the following sections, we first give a short overview of the related work. In Sect. III we introduce our exemplar-based parametric HMM movement representation. The action synthesis is described in Sect. IV. In Sect. IV-C we present the results of the robot experiments. In Sect. V we discuss our systematic evaluation of the precision of our action representation for recognition and synthesis. Conclusions in Sect. VI complete our paper.

II. Related Work

The discovery of mirror neurons motivated robotics researchers to look for action representations that can be used for both recognizing other agents' actions and generating the observer's own movements [3]. Among the representations that can be used both for synthesis and recognition: dynamic movement primitives [4], [5], recurrent neural network with parametric biases (RNNPB) [6], and hidden Markov models (HMMs) [7], [8], [9].

Hidden Markov models became popular in the context of speech recognition [10], [11]. One major advantage of HMMs is their ability to compensate for uncertainties in time. Inamura et al. [7] showed that by defining the observation states as postures on the human trajectories, HMMs can be used to represent specific movement trajectories, which they called proto-symbols. However, neither discrete nor continuous HMMs are able to generalize over a class of movements which vary according to a specific set of parameters (like for example reaching and pointing). One possibility to recognize an entire class of movements is to use a set of hidden Markov models (HMMs) in a mixture-of-experts approach, as first proposed in [12]. In order to deal with a large parameter space, one ends up with a lot of experts, and training becomes unsustainable.

As mentioned above, the extension of the classical HMMs into parametric HMMs was first proposed by Wilson and Bobick [1]. They developed an approach that is able to learn a parametric HMM based on a set of demonstrations, where training and recognition is performed by using the EM algorithm with parameterization parameters as latent variables. Note that this is different from [8], and [9], where HMMs are trained from multiple examples of an essentially same movement. However, Wilson and Bobick considered recognition only, more specifically, e. g., the recovery of the pointing direction based on wrist trajectories.

Contrary to Wilson and Bobick, we aim at recognition *and* synthesis of full arm movements for the control of humanoid robots within the same framework. In our work, "recognition" means to recognize the action itself as well as its parameterization. The synthesis of parametric actions implies the use of data of high dimension (stacked 3-D trajectories, one for each joint), and a high number of states for an accurate movement representation. It has already been shown both in robotics [13] and in computer graphics [14] that parametric blending of movements can result in physically feasible actions that can attain the goal of an action. These works, however, do not consider the problem of recognition.

Human motion capture [15] as such is not the topics of this paper. We note, however, that motion capture is an essential tool for the acquisition of training data for imitation. We experimented with magnetic systems, optical tracking devices, and general vision-based methods. General vision data was tested for recognition, but we used the data from a marker-based optical system as an initial input when working with the robot (see Fig. 1).

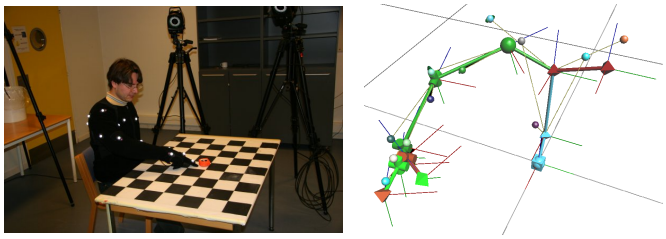


Fig. 1. *Left: Capturing Session* for our dataset. — *Right: Capture Model.* For motion capturing, the markers of the depicted model (tiny balls) are aligned to the captured markers (see left figure).

III. Movement Representation by PHMMs

In this section we first give a short introduction to HMMs and then discuss our extension to parametric HMMs.

A. Preliminaries of HMMs

A hidden Markov model is a finite state machine extended in a probabilistic manner. It is defined as a triple $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$. Let q_t be the hidden states of the model at time t and $\boldsymbol{x}$ the observations associated with each hidden state. Then, $\boldsymbol{B}$ defines the output distributions $b_i(\boldsymbol{x}) = \mathrm{P}(\boldsymbol{x}|q_t = i)$ of the hidden states. The transition matrix $\boldsymbol{A} = (a_{ij})$ defines the transition probability between the hidden states $i, j = 1, \ldots, N$, and thus encodes the temporal behavior of the modeled sequences. The initial state distribution is defined by the vector $\boldsymbol{\pi}$.

Our approach is based on continuous left-right HMMs [16]. The output probability distribution of each state i is modeled by a single Gaussian distribution $b_i(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. State transitions are either self-transitions or transitions to the successor. All other transition probabilities are set to zero. Given a sequence of postures $\boldsymbol{X} = \boldsymbol{x}_1 \ldots \boldsymbol{x}_t \ldots \boldsymbol{x}_T$ on an example trajectory, each Gaussian $\mathcal{N}_i(\boldsymbol{x}) := b_i(\boldsymbol{x})$ "covers" a section of the trajectory, where the state i increases over time. In the case of multiple trajectories, the Gaussians capture the variance of the training input. In addition, an HMM compensates for different progression rates of the training trajectories by varying the transition from one hidden state to the next. Obviously, the synthesis of movements is straightforward for this type of HMMs. For a comprehensive introduction to HMMs, we refer to [10], [11].

For recognition or classification, HMMs are generally used as follows: For each sequence class k, an HMM $\boldsymbol{\lambda}^k$ is trained by maximizing the likelihood function $\mathrm{P}(\mathcal{X}|\boldsymbol{\lambda}^k)$ with the Baum-Welch expectation maximization (EM) algorithm [11] over a given training data set $\mathcal{X}^k$. The classification of a specific output sequence $\boldsymbol{X} = \boldsymbol{x}_1 \ldots \boldsymbol{x}_T$ is done by identifying the class k for which the likelihood $\mathrm{P}(\boldsymbol{X}|\boldsymbol{\lambda}^k)$ is maximal. The forward-backward algorithm [11] is used to efficiently calculate these likelihoods.

One obvious approach to handling classes of parameterized actions for the purpose of parameter recognition is a mixture-of-experts approach [12] by sampling the parameter space. However, this approach suffers from the great number of HMMs needed to be trained and stored for all possible trajectories within one class. Therefore, we introduce the parameterization of an action as an additional model parameter. This is also the basic idea of [1].

B. Parametric HMM Framework

The basic idea of our approach to handling classes of parameterized actions is to generate a new HMM for novel action parameters by a locally linear interpolation of exemplar HMMs that were previously trained on exemplar movements with known parameters. The generation of an HMM $\boldsymbol{\lambda}^{\boldsymbol{\phi}}$ for a specific parameter $\boldsymbol{\phi}$ is carried out by component-wise linear

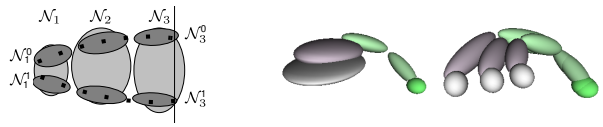


Fig. 2. *Left:* The upper three dark ellipsoids are depicting Gaussians $\mathcal{N}_1^0, \ldots, \mathcal{N}_3^0$ of states $i = 1, 2, 3$ of an local HMM $\boldsymbol{\lambda}^0$, whereas the lower three dark ellipsoids belong to a model $\boldsymbol{\lambda}^1$. The dots within the dark ellipsoids are sketching training sequences with parameterization $u = 0$, and $u = 1$ depending on the target at the vertical line. In addition, the Gaussians $\mathcal{N}_i$ of a global model $\boldsymbol{\lambda}$ are indicated in light gray. This global $\boldsymbol{\lambda}$ is a model for all training sequences with $u \in [0, 1]$. — *Right:* Similar to the left figure, some states of a global HMM (middle) and local HMMs (right) are shown. In contrast to the figure left, the HMMs are trained based on recorded 3D trajectories of a finger tip of a person pointing at three different positions at table-top (compare Fig. 1). The index finger starts always at the same point as modeled by the Gaussian which is sketched by the green ball, and approaches specific positions which are modeled by the light gray balls of the local HMMs. The global HMM used to setup these local HMMs has a disc like Gaussian, which models all approached positions at table-top.

interpolation of the nearby exemplar models. In the case of a single parameter u and two exemplar models $\boldsymbol{\lambda}^{u=0}$ and $\boldsymbol{\lambda}^{u=0}$, a new Gaussian $\mathcal{N}_i^u(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^u, \boldsymbol{\Sigma}_i^u)$ for a model $\boldsymbol{\lambda}^u$ with $u \in [0, 1]$ is generated by interpolation

$$\boldsymbol{\mu}_i^u = (1-u)\boldsymbol{\mu}_i^0 + u\boldsymbol{\mu}_i^1, \quad \boldsymbol{\Sigma}_i^u = (1-u)\boldsymbol{\Sigma}_i^0 + u\boldsymbol{\Sigma}_i^1. \quad (1)$$

This situation is described in Fig. 2. Such an approach, however, works *only if* two corresponding states of the two exemplar HMMs model the same semantical part of the trajectory (as in Fig. 2, left). Hence the state-wise alignment is vital and is described in Sec. III-B1. The expansion to the multi-variate case of parameterization $\boldsymbol{\phi}$ is then straightforward, e. g., by using bilinear ($\boldsymbol{\phi} = (u, v)$) or trilinear interpolation.

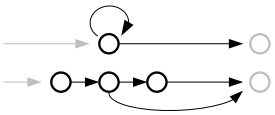


Fig. 3. *Time duration by State Replacement.* Each state is replaced by several (here, three) pseudo states which share the same output distribution.

1) Synchronization of HMMs: In this section we show how to assure that corresponding states of exemplar HMMs model the same semantical parts of movements. The required alignment of the states to the sequences is somehow similar to the alignment of two sequences by dynamic time warping (DTW) [17]. Here, however, we need to align the hidden states in the presence of many training sequences.

The underlying idea of the synchronization technique is to set up local exemplar HMMs $\boldsymbol{\lambda}^{\boldsymbol{\phi}}$ by using the invariance of HMMs to temporal variations. We proceed in two steps: firstly, a global HMM $\boldsymbol{\lambda}$ is trained based on the whole training set $\mathcal{X}$ of different parameterizations $\boldsymbol{\phi}$ of the same action. This is done using the EM algorithm mentioned in Sect. III-A. Such a global HMM is sketched in Fig. 2, left, by the light gray Gaussians. The situation that actions of different parameterizations are covered in such a symmetrical way as in Fig. 2 can be enforced by ensuring that the hidden state sequences pass the trajectory always in the sequential order from state 1 to N. This is done (a) by choosing left-right HMMs to model the movement, and (b) by allowing only state sequences that start in the first and end in the last state. In addition, (c) the invariance of the HMM to temporal variations, even though necessary, needs to be somehow constrained. We accomplish this by adding explicit time durations to the states of the HMM [11]. This is similar to constraining the warping in standard DTW approaches [17]. Here, by using explicit state durations, one can prevent that one state generates only one output for one sequence and a lot of outputs for another sequence. Otherwise, one would end up with an improper alignment of the sequences (see [17], Fig. 2 C, for an improper, and B for a proper alignment). To avoid the numerical problems associated with scaling [11], we replaced each state of the left-right HMM with pseudo states that share one Gaussian (see Fig. 3). This forces the hidden states sequences to stay in a state, e. g., as in Fig. 3, for at least two and for maximal three time steps.

In the second step, we consider the partial training set $\mathcal{X}^{\boldsymbol{\phi}}$ associated with a specific parameterization $\boldsymbol{\phi}$. Using this data set, we estimate the local HMM $\boldsymbol{\lambda}^{\boldsymbol{\phi}}$ again by using the EM algorithm, but now, we use the parameters of the global HMM $\boldsymbol{\lambda}$ as an initial value for the EM algorithm. In addition, to preserve the state alignment of the local HMM as it is given by the global HMM, we fix the means after the first EM step. In the following, we will exemplify, that this adapted EM procedure gives a proper state alignment of the local HMMs: In the first E step of the EM algorithm, the posterior probabilities $\gamma_t^k(i) = \mathrm{P}(q_t = i|\boldsymbol{X}^k, \boldsymbol{\lambda})$ of being in state i at time t are computed for each sequence $\boldsymbol{X}^k = \boldsymbol{x}_1^k \ldots \boldsymbol{x}_T^k$ of the training set $\mathcal{X}^{\boldsymbol{\phi}}$. This is done based on the current parameter values, which are at this point the values of the global HMM. Thus, $\gamma_t^k(i)$ is the "responsibility" of state i for generating $\boldsymbol{x}_t^k$, as given by the global HMM. In the M step of the EM algorithm, each $\boldsymbol{\mu}_i$ of a Gaussian of state i is re-estimated as $\gamma_t^k(i)$-weighted mean:

$$\boldsymbol{\mu}_i = \frac{\sum_{t,k} \gamma_t^k(i) \boldsymbol{x}_t^k}{\sum_{t,k} \gamma_t^k(i)} \quad (2)$$

Now, consider Fig. 2. The responsibilities $\gamma_t^0(i)$ of the upper sequence $\boldsymbol{x}_1^0 \boldsymbol{x}_2^0 \ldots \boldsymbol{x}_7^0$ for $t = 1, 2$ are large for $i = 1$ but small for $i > 1$ under consideration of the position of the Gaussian $\mathcal{N}_1$ of the global HMM $\boldsymbol{\lambda}$. Thus, $\boldsymbol{\mu}_1^0$ of $\mathcal{N}_1^0$ of the local HMM $\boldsymbol{\lambda}^0$, as calculated by Eq. (2), lies as desired between $\boldsymbol{x}_1^0$ and $\boldsymbol{x}_2^0$. Similarly, $\boldsymbol{\mu}_1^1$ of the local HMM $\boldsymbol{\lambda}^1$ computed based on the lower sequence $\boldsymbol{x}_1^1 \boldsymbol{x}_2^1 \ldots$ would lie between $\boldsymbol{x}_1^1$ and $\boldsymbol{x}_2^1$. Hence, the alignment of $\boldsymbol{\mu}_1^0$ and $\boldsymbol{\mu}_1^1$ of the local HMMs $\boldsymbol{\lambda}^0$ and $\boldsymbol{\lambda}^1$ as given by the global HMM $\boldsymbol{\lambda}$ is ensured.

2) Synthesis, Recognition, and Parameters: At first we consider the synthesis in the general case of a parametric movement parameterized by $\boldsymbol{\phi} \in \mathbb{R}^d$. Let $\boldsymbol{\phi}$ be the parameterization of the movement that needs to be generated. For synthesis we need properly synchronized HMMs $\boldsymbol{\lambda}^{\boldsymbol{\phi}_n}$ trained for movements with parameterization $\boldsymbol{\phi}_n$ for the 2^d corners $\boldsymbol{\phi}_n$ of a d-dimensional cuboid (or at least a warped version of

such an cuboid) that contains the required parameterization $\boldsymbol{\phi}$. Since $\boldsymbol{\phi}$ is inside the cuboid, there exist interpolation parameters $\boldsymbol{\omega}(\boldsymbol{\phi})$ such that $\boldsymbol{\phi}$ can be expressed as a d-linear combination of $\boldsymbol{\phi}_n$ with parameters between 0 and 1. Lets denote by $\boldsymbol{\lambda}^{\boldsymbol{\phi}}$ the component wise d-linear interpolation of the HMMs $\boldsymbol{\lambda}^{\boldsymbol{\phi}_n}$ interpolated with interpolation parameters $\boldsymbol{\omega}(\boldsymbol{\phi})$. This $\boldsymbol{\lambda}^{\boldsymbol{\phi}}$, based on the local HMMs $\boldsymbol{\lambda}^{\boldsymbol{\phi}_n}$, is what we call a parametric HMM (PHMM). The calculation of the parameters $\boldsymbol{\omega} = (u, v, w)$ in the case of 3-D, or (u, v) in the case of 2-D parameterization interpolation, which we used in the experiments, is straight-forward, see Sect. IV. The bilinear interpolation formula is given by (5), the trilinear by (4), the extension to the d-linear case can be constructed easily. The movement trajectory $\boldsymbol{f}(t)$ for a specific parameterization $\boldsymbol{\phi}$ can be synthesized using, e. g., linear spline interpolation of the sequence of means $\boldsymbol{\mu}_1^{\boldsymbol{\phi}}\boldsymbol{\mu}_2^{\boldsymbol{\phi}}\boldsymbol{\mu}_3^{\boldsymbol{\phi}}\ldots$ of $\boldsymbol{\lambda}^{\boldsymbol{\phi}}$ with respect to the expected time durations encoded in the transition matrix of $\boldsymbol{A}^{\boldsymbol{\phi}}$.

The recognition of the type of movement and its parameterization can be accomplished as follows. For each class k we have a PHMM $\boldsymbol{\lambda}_k^{\boldsymbol{\phi}}$ that represents a parametric movement. Now, lets assume that we need to classify a sequence $\boldsymbol{X}$. We start by estimating the most likely parameter $\boldsymbol{\phi}_k$ for each possible parameterized action class k. This involves maximizing the likelihood functions:

$$\boldsymbol{\phi}_k = \underset{\boldsymbol{\phi}\in[0-\varepsilon,1+\varepsilon]^d}{\arg\max}\ \mathrm{P}(\boldsymbol{X}|\boldsymbol{\lambda}_k^{\boldsymbol{\phi}}), \tag{3}$$

The class identity is given by the class for which the likelihood $\mathrm{P}(\boldsymbol{X}|\boldsymbol{\lambda}_k^{\boldsymbol{\phi}})$ is the highest. In addition, the associated parameter $\boldsymbol{\phi}_k$ gives the most likely parameterization. In the experiments we maximized the log likelihood functions $l_k(\boldsymbol{\phi}) = \log \mathrm{P}(\boldsymbol{X}|\boldsymbol{\lambda}_k^{\boldsymbol{\phi}})$ (see Fig. 9 (c)). This is done using the gradient descent method and numerical derivatives of $l_k(\boldsymbol{\phi})$.

In our evaluation of the movement representation, the parameterization is given by 2-D coordinates in the plane of a table-top. In that experiment we have up to 3×3 local HMMs for each movement type which form a regular raster. In this case the first guess is always based on the HMMs defined at the outermost positions, then the estimate is refined based on the HMMs with parameter positions that define the smallest rectangle which includes the first guess.

IV. Transfer of Movements to the Robot

In this section we consider how to generate arbitrary robot reaching and pointing movements using PHMMs. Our approach can be easily used for other types of parametric movements. For training we use example reaching and pointing movements that stretch out or point to a number of different grasping positions distributed in the robot workspace. An example of the workspace is shown in Fig. 4, where the eight target $\boldsymbol{t}_{ijk}$ positions form a cuboid. This way, the location of the gripper on the table as well as up to the certain height above the table can be controlled. The training movements are used as explained in III to train the PHMM. The PHMM is based on eight local HMMs, one for each of the eight

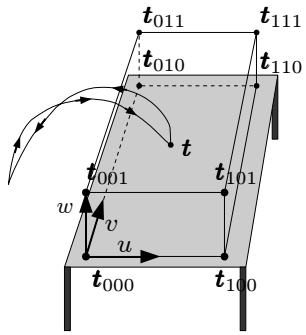


Fig. 4. *Target Points of Movements* at a table-top. In addition, a finger trajectory of a pointing movement with target point $\boldsymbol{t}$ is shown. The coordinates u, v, w are used for interpolation.

example positions. Note that at each training position, several movements are used to train the HMM. The training step assures that the HMMs are all synchronized for interpolation.

In our experiment, the recorded reaching and pointing movements start and stop at the same base position (hand beside hip), i. e. we recorded both the reaching for the object or pointing movement and the withdrawing movement. In order to be able to synthesize realistically looking sequences of several pointing and reaching movements without discontinuities if played back in sequence, the base pose at the beginning and at the end of the motions is normalized. This is done by blending the trajectories with respect to the mean starting position.

A. Synthesis of Robot Movements for Specific Positions

For each movement type we have a PHMM of eight local HMMs trained for specific movements to the targets $\boldsymbol{t}_{ijk}$, like shown in Fig. 4. The recorded movements are a stacked vector $\boldsymbol{x} = (\boldsymbol{p}^\top, \boldsymbol{q}^\top, \ldots)^\top$ of the trajectories of the right arm shoulder $\boldsymbol{p}$, elbow $\boldsymbol{q}$, thumb, finger, and its knuckle.

Since we consider the trajectories relative to the shoulder position we calculate the mean shoulder position over all example trajectories and use this as reference. Now, consider an arbitrary target $\boldsymbol{t} = \boldsymbol{t}^{uvw}$ in the workspace that is given by interpolating the corners $\boldsymbol{t}_{ijk}$ by trilinear interpolation with the parameters (u, v, w):

$$\boldsymbol{t}^{uvw} = \bar{w}\boldsymbol{t}^{uv0} + w\boldsymbol{t}^{uv1} \tag{4}$$

$$\boldsymbol{t}^{uv0} = \bar{v}(\bar{u}\boldsymbol{t}_{000} + u\boldsymbol{t}_{100}) + v(\bar{u}\boldsymbol{t}_{010} + u\boldsymbol{t}_{110}) \tag{5}$$

$$\boldsymbol{t}^{uv1} = \bar{v}(\bar{u}\boldsymbol{t}_{001} + u\boldsymbol{t}_{101}) + v(\bar{u}t_{011} + t\boldsymbol{t}_{111}), \tag{6}$$

where

$$\bar{w} = 1 - w, \quad \bar{v} = 1 - v, \quad \bar{u} = 1 - u.$$

Then a movement $\boldsymbol{f} = (\boldsymbol{p}^\top, \boldsymbol{q}^\top, \ldots)^\top$ for the target position $\boldsymbol{t}$ can be synthesized as described in Sect. III-B2. Essentially, that is interpolating each component (shoulder, elbow, ...) of the means of the corresponding states of the local HMMs as given by the interpolation formula (4). The interpolation

parameters (u, v, w) for a specific target $\boldsymbol{t}$ are easily calculated by

$$u = \frac{\boldsymbol{t} - \boldsymbol{t}_{000}}{|\boldsymbol{t} - \boldsymbol{t}_{000}|} \cdot \frac{\boldsymbol{t}_{100} - \boldsymbol{t}_{000}}{|\boldsymbol{t}_{100} - \boldsymbol{t}_{000}|}, \tag{7}$$

$$v = \frac{\boldsymbol{t} - \boldsymbol{t}_{000}}{|\boldsymbol{t} - \boldsymbol{t}_{000}|} \cdot \frac{\boldsymbol{t}_{010} - \boldsymbol{t}_{000}}{|\boldsymbol{t}_{010} - \boldsymbol{t}_{000}|}, \tag{8}$$

$$w = \frac{\boldsymbol{t} - \boldsymbol{t}_{000}}{|\boldsymbol{t} - \boldsymbol{t}_{000}|} \cdot \frac{\boldsymbol{t}_{001} - \boldsymbol{t}_{000}}{|\boldsymbol{t}_{001} - \boldsymbol{t}_{000}|}. \tag{9}$$

Since the robot is smaller than the demonstrator, these trajectories are scaled to fit to the overall arm length of the robot. Of course the distances, e. g. between the wrist and elbow, are not preserved by interpolation, but become slightly shorter or longer than the true limb lengths. However, this does not matter if the movements are transferred to the robot as described in Sect. IV-B.

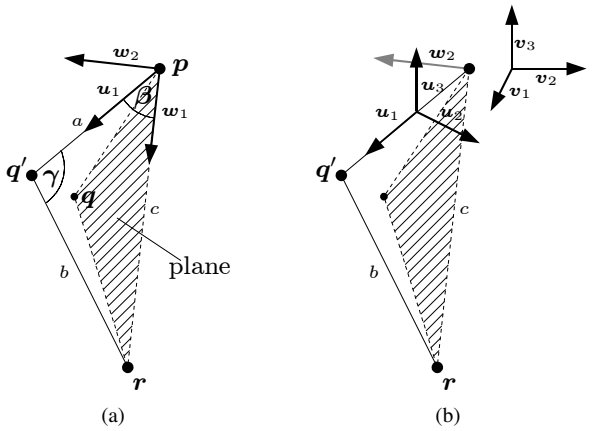


Fig. 5. In figure (a), the points $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{r}$, which represent the shoulder, elbow, and hand, define a plane, on which the elbow $\boldsymbol{q}'$ of the robot should lie. Figure (b) shows the two reference coordinate systems $(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3)$ and $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$. The coordinate system $(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3)$ is the reference of the upper arm, whereas $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ is fixed and does not move with the upper arm.

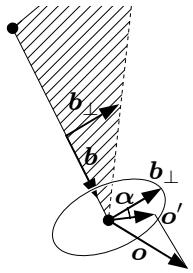


Fig. 6. *Rotation of the Robot's Wrist*. The twist angle $\alpha = \measuredangle(\boldsymbol{b}_\perp, \boldsymbol{o}')$ of the wrist is calculated in the twist plane.

B. Calculation of Joint Angles

The interpolation of Sect. IV-A results in an arm trajectory which is described by the Cartesian trajectories of the shoulder, elbow, wrist, thumb, index-finger and its knuckle. In this section, we will describe how the robot joint angles of the upper arm are computed from this data. The joints under consideration are the shoulder joint with three degrees of freedom, and the elbow joint with one degree of freedom. These values are determined only on the basis of the human elbow and hand position relative to the shoulder. The hand position $\boldsymbol{r}$ (Fig. 5) is given as the mean of the human thumb, index finger, and knuckle of that finger. Furthermore, the orientation of the hand is given by the thumb and the finger.

To account for a difference in size between the human demonstrator and the robot, human data is scaled to the size of the robot as described in Sect. IV-A. For the calculation of joint angles, only the relative positions of the elbow $\boldsymbol{q}$ and hand $\boldsymbol{r}$ with respect to the shoulder $\boldsymbol{p}$ are of interest. The shoulder can therefore be assumed to be at the origin ($\boldsymbol{p} = \boldsymbol{0}$); and the human hand and elbow positions, $\boldsymbol{r}$ and $\boldsymbol{q}$, of Fig. 5 (b) are appropriately scaled for the robot.

The four joint angles of the robot arm are calculated so that the robot hand is at the same position as the scaled human hand, and that the elbow $\boldsymbol{q}'$ is in the plane defined by three points $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{r}$ of the human demonstrator. The elbow position $\boldsymbol{q}'$ of the robot differs from the (scaled) elbow position of the demonstrator if the proportions of the robot, i. e. the upper arm compared to the forearm, differ from those of the demonstrator.

To calculate the shoulder angles (see Fig. 5 (a)), the direction $\boldsymbol{u}_1$ of the elbow $\boldsymbol{q}'$ is required. The angle γ of the elbow is defined by the length a of the upper arm, the length b of the lower arm (elbow – hand) of the robot, and the distance $c = \overline{\boldsymbol{pr}}$ from the shoulder $\boldsymbol{p}$ to the target point $\boldsymbol{r}$. We calculate γ as

$$\gamma = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right), \tag{10}$$

and

$$\beta = \arccos\left(\frac{a^2 + c^2 - b^2}{2ac}\right). \tag{11}$$

Now, $\boldsymbol{u}_1$ is given by

$$\boldsymbol{u}_1 = \cos\beta \cdot \boldsymbol{w}_1 \; + \; \sin\beta \cdot \boldsymbol{w}_2, \tag{12}$$

where

$$\boldsymbol{w}_1 = \overrightarrow{\boldsymbol{pr}} \,/\, \overline{\boldsymbol{pr}} \tag{13}$$

$$\boldsymbol{w}_2 = \boldsymbol{w}_2' \,/\, |\boldsymbol{w}_2'| \tag{14}$$

$$\boldsymbol{w}_2' = \overrightarrow{\boldsymbol{pq}} - \langle \overrightarrow{\boldsymbol{pq}}, \boldsymbol{w}_1 \rangle \, \boldsymbol{w}_1. \tag{15}$$

For HOAP-3, the three shoulder joint angles can be calculated as Cardan angles for the rotation between the coordinate systems $\{\boldsymbol{v}_i\}$ and $\{\boldsymbol{u}_i\}$ (see Fig. 5 (b)). The Cardan angles[1] ϕ, θ, ψ are calculated using the rotational matrix between the

[1]Cardan angles ϕ, θ, ψ define a rotation $\boldsymbol{R}$ as $\boldsymbol{R} = \boldsymbol{R}_\psi^{\boldsymbol{z}} \boldsymbol{R}_\theta^{\boldsymbol{y}} \boldsymbol{R}_\phi^{\boldsymbol{x}}$.

coordinate systems[2]:

$$\phi = \arctan2(r_{32}, r_{33}) \tag{16}$$

$$\theta = -\arcsin(r_{31}) \tag{17}$$

$$\psi = \arctan2(r_{21}, r_{11}) \tag{18}$$

where

$$\boldsymbol{R} = (r_{ij}) \;=\; [\boldsymbol{u}_1|\boldsymbol{u}_2|\boldsymbol{u}_3]^\top\,[\boldsymbol{v}_1|\boldsymbol{v}_2|\boldsymbol{v}_3]\,. \tag{19}$$

The still unknown vectors $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$ are given by the equations:

$$\boldsymbol{u}_2 = \boldsymbol{u}_2' \,/\, |\boldsymbol{u}_2'| \tag{20}$$

$$\boldsymbol{u}_2' = -\boldsymbol{w}_2 - \langle -\boldsymbol{w}_2, \boldsymbol{u}_1\rangle\,\boldsymbol{u}_1 \tag{21}$$

$$\boldsymbol{u}_3 = \boldsymbol{u}_1 \times \boldsymbol{u}_2. \tag{22}$$

The orientation of the hand was initially extracted from the motion data based on the direction given by the vector from the finger to thumb. However, it turns out to be better to align the robot gripper to the plane of the table, at least for that part of the motion which is used to interact with the objects. The degree of opening or closing the robot gripper was initially set to the distance between the finger and the thumb of the human. However, it turns out to be better to use maximal gripper opening while interacting with an object, as we used objects of different shapes in the recording session.

Since the wrist of the HOAP-3 has only one degree of freedom (twist), only the projection of the orientational direction vector $\boldsymbol{o}$ (finger⟶thumb) onto the rotation plane of the wrist is used to set its twist angle. This is illustrated in Fig. 6. The joint angle is given by (see Fig. 6):

$$\alpha = \arccos\left(\frac{\langle \boldsymbol{b}_\perp, \boldsymbol{o}'\rangle}{|\boldsymbol{b}_\perp|\,|\boldsymbol{o}'|}\right) \cdot \mathrm{sign}(\langle \boldsymbol{o}', \boldsymbol{b} \times \boldsymbol{b}_\perp\rangle), \tag{23}$$

where

$$\boldsymbol{o}' = \boldsymbol{o} - \frac{\langle \boldsymbol{o}, \boldsymbol{b}_\perp\rangle}{|\boldsymbol{b}_\perp|^2}\boldsymbol{b}_\perp. \tag{24}$$

C. Experiments with a Humanoid Robot HOAP-3

We tested the effectiveness of our approach by implementing a task involving a number of objects that are first associated with different openings on the table and need to be placed into the correct opening (see also Fig. 7 (left)). An online demo is available via web page [2]; the recording of demonstrated movements is not included. The experiment proceeds in three phases:

- The PHMMs for reaching and pointing movements are learnt.
- Human demonstrator shows to the robot which object belongs to which hole. The robot associates each object with the appropriate hole.
- Afterwards, the objects are placed again at arbitrary positions. In this phase, the human points at one of the object, which is then identified and placed into the associated hole using the previously learnt PHMMs. This is repeated until all objects are removed.

To technically implement the placement of an object into the appropriate hole, the robot first estimates the transformation between the robot and the camera coordinate system by moving the hand of the robot to at least four different configurations, which is enough to calculate the transformation between two camera systems. After the object is identified, the robot reaches for the object using the learnt PHMM. We use here only a part of the interpolated movement corresponding to the motion before the grasp. To be able to grasp the object, the robot reaches towards the position in front of the object (see Fig. 8). Relocating the object from the current position on the table to the hole is accomplished by generating a sequence of reaching positions on the table and using the PHMM to generate the corresponding reaching movements for these positions. The configurations where the robot reaches for these positions are extracted and interpolated to generate the relocation trajectory. In this way we ensure that the robot uses only natural arm postures. Grasping and releasing of the object is implemented using standard robotics methods. The withdrawing movement is realized in the same way as the reaching movement, the only difference being that in this case we use the part of the trajectory after the grasp.

V. Evaluation of Movement Representation

In the evaluation we focus our considerations on pointing (see Fig. 1) and reaching actions, which are not only the most important movements in our behavioral experiment but are also important in other interaction scenarios. Both are performed in a very similar way, starting and ending in the same base position (arm along the body). The motion data of our systematic evaluation is acquired using an eight camera Vicon system with cameras running at 60 Hz (see Fig. 1). The recognition and synthesis experiments are based on seven 3-D points located at different segments of a human body. The seven data points are: sternum; shoulder and elbow of the right arm; knuckles, index finger, and thumb of the right hand.

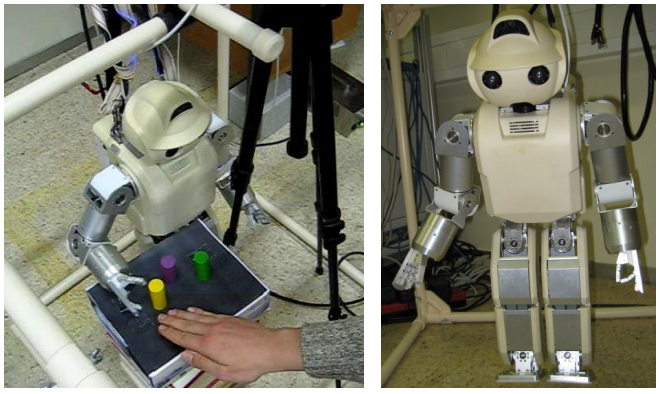


Fig. 7. *Our Experimental Setup.* A person advises the robot HOAP-3 how to clean up objects. *Online Demo*, available via web page [2].

[2]Contrary to $\arctan(a/b)$, $\arctan2(a,b) \in (-\pi, \pi]$ respects the signs of a and b.

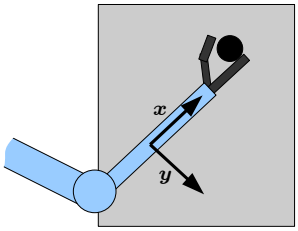


Fig. 8. *Approach Motion* along the x-axis to grasp an object

The exemplar positions at table-top form a regular raster, which covers a region of 80cm × 30cm (width × deeps). For training, a 3×3 raster is used, where 10 repetitions have been recorded for each exemplar position and each action type (pointing, reaching). For evaluation, a 5×7 test-raster is used with 4 repetitions for each raster position to allow a good evaluation statistic. All in all, for testing we used several hundreds of movements.

A. Training: Setup of PHMMs

Training and setup of the exemplar PHMMs for grasping and pointing is done as described in Sec. III-B1. We used PHMMs of 20 states, where the hidden state sequences are forced to stay between 4 and 6 time steps in each state. The training sequences are normalized to 100 samples. We train the PHMMs based on data of the full 3×3 raster (9 exemplar HMMs) or based on a 2×2 raster, which consists of the four outermost exemplar positions of the 3×3 raster at table-top. These PHMMs will be refered in the following as 3×3 or 2×2 PHMM of reaching or pointing.

B. Synthesis

Synthesis is done as described in Sec. III-B2 with the setup described in the section above. The performance of synthesis is systematically evaluated by plotting the synthesis error for each of the positions of the 5×7 test-raster. Therefore, the error is based on the distance between each synthesized movement and an average of the four test exemplars of a test-raster positions. The averaging is done by training and re-synthesizing from an 80 state HMM. Again, the re-synthesis is a function $\bar{\boldsymbol{f}}(t) = (\bar{\boldsymbol{f}}_i(i))_{i=1}^{7}$ of "stacked" 3-D trajectories $\boldsymbol{f}_i(t)$ (elbow, wrist, ...). The final error ε is calculated as the root-mean-square error between the time warped synthesis, $\boldsymbol{f}(t)$, and re-synthesized average, $\bar{\boldsymbol{f}}(t)$:

$$\varepsilon = \sqrt{\int \sum_{i=1}^{7} \frac{(\boldsymbol{f}_i(\alpha(t)) - \bar{\boldsymbol{f}}_i(\bar{\alpha}(t)))^2}{7} \mathrm{d}t \Big/ \int \alpha(t) dt}, \quad (25)$$

where $\alpha(t)$ and $\bar{\alpha}(t)$ are warping functions. As the starting and ending points of the reference $\bar{\boldsymbol{f}}(t)$ sometimes vary slightly, the first and last 10% of the sequences are not considered. Obviously, the error ε is normalized w. r. t. the length of the sequence.

The Fig. 9 (a), and (b) compare the synthesis errors of 2×2 and 3×3 exemplar PHMMs. Clearly, the performance in the middle of the covered region increases, if the 3×3 PHMM is used. Fig. 9 (c), and (d) show the fact that the results for the reaching action are very similar to the pointing actions. If the outer regions are neglected the synthesis errors are approximately 1.8cm both in the case of reaching and pointing.

C. Recognition

For recognition, we evaluate (a) the performance of estimating the parameter of a movement, (b) the rate of the correct classifications of movement types, and (c) the robustness to noise.

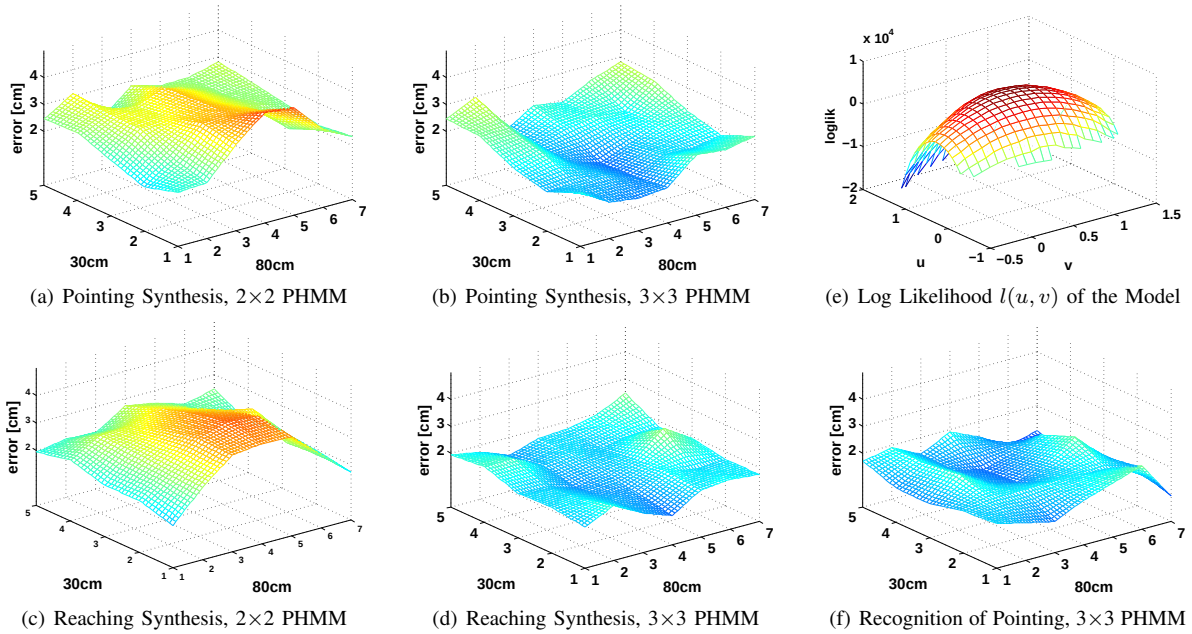
First it is worth to take a look at Fig. 9 (e), which shows that the optimization problem of maximizing the log likelihood function $l(u, v) = \log P(\boldsymbol{X}|\boldsymbol{\lambda}^{uv})$ given a movement is tractable by standard optimization techniques (smoothness and strict concavity). Thus, the most likely parameterization (u, v) can easily be estimated. The errors for each position of the 5×7 test-raster are calculated as the average deviation of the estimated position and the ground truth position for the test example movements. The estimation accuracy of the table-top positions behaves very similar to the results of the synthesis Fig. 9 (f). The performance increases similar to the synthesis in the inner region for our 3×3 PHMM compared to the 2×2 PHMM (not depicted). The rate of correctly classified types of the 280 grasping and pointing test movements is 94% for the 2×2 PHMMs, and changes just insignificantly for the 3×3 PHMMs.

We tested the robustness of estimating the parameters of movements by adding Gaussian noise to each component of the samples of the movements. Here, we realized no significant influence for independently distributed noise with $\sigma < 15$cm. Obviously, that is due to a large number of samples in the sequence.

VI. Conclusions

We have presented a novel approach to represent, classify, and imitate parametric movements using parametric hidden Markov models. Our approach contains several contributions: (a) how to learn and represent parametric human movements, (b) how to use this representation for action recognition, (c) how to transform and project the actions onto the embodiment of the robot, and (d), how to generate the actions on a new embodiment. Furthermore, we have solved several subproblems such as multi-dimensional time warping of the multiple training sequences so that HMMs can be properly interpolated.

We systematically evaluated the synthesis and recognition performance of the proposed PHMM framework. The experiments show the accuracy of our approach for the generation of new movements and for the estimation of the associated movement parameters (errors of ≈2cm). This shows that the newly generated movements are similar to the observed movements. This also was confirmed in the behavioral experiment of Sec. IV-C, where the generated reaching movements were similar to the training examples, and, like the examples, avoided collisions with the table. This could not be guaranteed if the movement was generated by standard robotics approaches. It is



(a) Pointing Synthesis, 2×2 PHMM (b) Pointing Synthesis, 3×3 PHMM (e) Log Likelihood $l(u, v)$ of the Model

(c) Reaching Synthesis, 2×2 PHMM (d) Reaching Synthesis, 3×3 PHMM (f) Recognition of Pointing, 3×3 PHMM

Fig. 9. *Error of Synthesis* (a-d) *and Recognition* (f): evaluated at 7×5 raster positions for different PHMMs resolutions, a plot of the log likelihood for the PHMM parameters (u, v) is given in (e).

worth to note that the synthesis error does not affect the robot-object interaction since we interpolate based on the gripper position which is encoded in the movements.

The classification rate of the type movement is $\approx 94\%$. It is worth pointing out that this recognition rate is achieved without any kind of diagnostic features. Furthermore, it should be noticed that the movements of pointing and reaching are very similar. In earlier experiments we had trained classical HMMs on pointing and reaching actions where the training movements were directed similarly (up to the natural variance of human performances). During the tests, our classical HMMs reached a recognition rate of $\approx 85\%$. The fact that the PHMMs lead to considerably better recognition rates shows that they are much better in describing the actions and in compensating for natural variability of the performances.

We conclude that PHHMs are suitable for imitation because they are both generative and accurate for recognition.

ACKNOWLEDGMENT

This work was partially supported by EU through grant PACO-PLUS, FP6-2004-IST-4-27657.

REFERENCES

[1] A. D. Wilson and A. F. Bobick, "Parametric hidden Markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.

[2] D. Herzog, A. Ude, and V. Krueger, "Hoap-3 online demo." [Online]. Available: http://www.cns.atr.jp/~aude/rules.mpg

[3] V. Krüger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: A review on action recognition and mapping," *Advanced Robotics*, vol. 21, no. 13, pp. 1473–1501, 2007.

[4] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, Mass.: MIT Press, 2003, pp. 1547–1554.

[5] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics Research: The Eleventh International Symposium*, P. Dario and R. Chatila, Eds. Berlin, Heidelberg: Springer, 2005, pp. 561–572.

[6] J. Tani, M. Ito, and Y. Sugita, "Self-organization of distributedly represented multiple behavior schemata in a mirror system: Reviews of robot experiments using RNNPB," *Neural Networks*, vol. 17, no. 8-9, pp. 1273–1289, 2004.

[7] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *Int. J. Robotics Research*, vol. 23, no. 4-5, pp. 363–377, 2004.

[8] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, pp. 370–384, 2006.

[9] T. Asfour, F. Gyarfas, P. Azad, and R. Dilmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Genoa, Italy, December 2006, pp. 40–47.

[10] X. Huang, Y. Ariki, and M. Jack, *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[11] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, pp. 4–15, January 1986.

[12] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[13] A. Ude, M. Riley, B. Nemec, A. Kos, T. Asfour, and G. Cheng, "Goal-directed action synthesis from a library of example movements," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Pittsburgh, Pennsylvania, December 2007.

[14] A. Safonova and J. Hodgins, "Analyzing the physical correctness of interpolated human motion," in *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2005, pp. 171–180.

[15] T. Moeslund, A. Hilton, and V. Krueger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–127, 2006.

[16] S. Gunter and H. Bunke, "Optimizing the number of states, training iterations and Gaussians in an HMM-based handwritten word recognizer," *icdar*, vol. 01, p. 472, 2003.

[17] E. Keogh and M. Pazzani, "Derivative dynamic time warping," in *Proceedings of the 1st SIAM International Conference on Data Mining (SDM'2001), Chicago, USA*, 2001.

Parametric Hidden Markov Models for Recognition and Synthesis of Movements

Dennis Herzog, Volker Krüger, Daniel Grest
Aalborg University Copenhagen
Lautrupvang 15, 2750 Ballerup, Denmark
{deh,vok,dag}@cvmi.aau.dk

Abstract

In humanoid robotics, the recognition and synthesis of parametric movements plays an extraordinary role for robot human interaction. Such a parametric movement is a movement of a particular type (semantic), for example, similar pointing movements performed at different table-top positions.

For understanding the whole meaning of a movement of a human, the recognition of its type, likewise its parameterization are important. Only both together convey the whole meaning. Vice versa, for mimicry, the synthesis of movements for the motor control of a robot needs to be parameterized, e.g., by the relative position a grasping action is performed at. For both cases, synthesis and recognition, only parametric approaches are meaningful as it is not feasible to store, or acquire all possible trajectories.

In this paper, we use hidden Markov models (HMMs) extended in an exemplar-based parametric way (PHMM) to represent parametric movements. As HMMs are generative, they are well suited for synthesis as well as for recognition. Synthesis and recognition are carried out through interpolation of exemplar movements to generalize over the parameterization of a movement class.

In the evaluation of the approach we concentrate on a systematical validation for two parametric movements, grasping and pointing. Even though the movements are very similar in appearance our approach is able to distinguish the two movement types reasonable well. In further experiments, we show the applicability for online recognition based on very noisy 3D tracking data. The use of a parametric representation of movements is shown in a robot demo, where a robot removes objects from a table as demonstrated by an advisor. The synthesis for motor control is performed for arbitrary table-top positions.

1 Introduction

For the design of humanoid robots, the synthesis and recognition of humanlike movements plays an extraordinary role, as emphasized in [2]. On the one hand, it is desirable for the robot to synthesize movements in a humanlike way. On the other hand, the robot needs be able to recognize human movements. For recognition, mirror neurons, which are supposed to map movements of an observed person onto ones own embodiment, could

justify a generative approach, like HMMs. On the recognition side, it is necessary to recognize the movement type, as well as its parameterization. Only both together convey the whole semantics, e.g., of "pointing at *this* specific object" (see Fig. 1).

Beside the field of robotics, synthesis concerns 3D human body tracking. In human body tracking, one is interested in using motion models in order to constrain the parameter space (e.g., for simple cyclic motions [7]). In both cases, one is interested in teaching the system in an easy and efficient manner an additional parametric movement, such that the demonstration of a sparse set of exemplars of different movement parameterizations enables the system to synthesize the movement for arbitrary parameterizations. In case of a humanoid robot, the synthesis should then allow the robot to perform the learned grasping movements with new parameterizations, e.g., grasping objects at arbitrary positions. In case of the 3D body tracking, synthesis would allow a better pose prediction, and even allows an estimate of parametric actions instead of the full joint configuration.

Most current approaches model movements with a set of movement *prototypes*, and identify a movement by identifying the prototype which explains the observed movement best. This approach, however, has its limits concerning efficiency when the space of possible parameterizations is large. Another approach is the use of diagnostic features, e.g., the distance from chest to arm. Such an approach might perform well in the case of movements leading to specific locations, but are doubtful in the cases of parametric movements.

A pioneering work in this context was done by Wilson and Bobick [10]. They presented a parametric HMM approach that is able to learn an HMM based on a set of demonstrations, where training and recognition is performed by the EM algorithm, where the parameterizations of a movement are taken as latent variables. They mainly aim at recognition, e.g., like recovering the pointing directions based on wrist trajectories, or like the occurrence of different kind of gestures.

In this paper, we use a similar parametric model. Contrary to Wilson and Bobick, we aim at recognition as well as synthesis of full arm movements. Here, recognition means to classify the movement type, and to recover the parameterization of the movement. The synthesis implies the use of data of high dimension (stacked trajectories), and a high number of states for an accurate movement representation. This complicates the training of the model. However, in the case of smooth trajectories also a smaller number of state might be sufficient in combination with spline interpolation. As synthesis and recognition is carried out through linear interpolation, a proper alignment of exemplar movements with different parameterizations is essential. We handle this by constraining the time

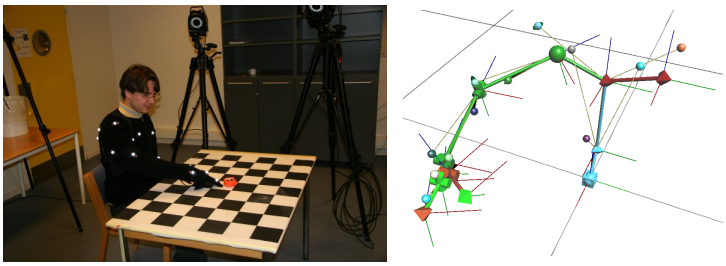


Figure 1: *Left:* Capturing Session for Our Dataset. — *Right:* Capture Model. For motion capturing, model markers (tiny balls) are aligned to captured markers (left figure).

warping capability of the HMMs.

In the experiments we focus on a systematical evaluation of the parameterization of movements. Therefore, we consider grasping movements (reaching out for an object to grasp), and a pointing at movement. We consider these movements as most important for the most scenarios of robot human interaction. Both movements have very similar trajectories (starting, and ending in the same base pose, and the object is released directly after the grasp). Thus, a simple diagnostic feature like arm velocity, or the distance from hand to chest would fail.

In the following section, we give a short overview of the related work. In Sec. 3 we provide some basics to introduce our exemplar-based parametric HMM in Sec. 4. Experimental results of Sec. 5 contain an extensive and systematic evaluation of the synthesis and recognition capability of our model, and an online demo for recognition, which shows the online applicability and robustness for 3D tracking in the presence of noise. In a robot demo, we show the use of our approach on a humanoid robot. Conclusions in Sec. 6 complete our paper.

2 Related Work

Most approaches for movement representation that are of interest in our problem context are trajectory based: Training trajectories, e.g., sequences of human body poses, are encoded in a suitable manner. Newly incoming trajectories are then compared with the previously trained ones. A recent review can be found in [6].

Some of the most common approaches to represent movement trajectories use hidden Markov models (HMMs) [3, 8]. HMMs offer a statistical framework for representing and recognition of movements. One major advantage of HMMs is their ability to compensate for some uncertainty in time. However, due to their nature, general HMMs are only able to model specific movement trajectories, but they are not able to generalize over a class of movements that vary accordingly to a specific set of parameters. One possibility to recognize an entire class of movements is to use a set of hidden Markov models (HMMs) in a mixture-of-experts approach, as first proposed in [4]. In order to deal with a large parameter space one ends up with a lot of experts, and training becomes un-sustainable.

Another extension of the classical HMMs into parametric HMMs was presented in [10], as mentioned above. A more recent approach was presented by [1]. In this work, the interpolation is carried out in spline space where the trajectory of the end-effector is modeled. Apart from the fact that the authors have not yet performed an evaluation of their system, their approach does not seem suitable for controlling entire arm movements.

In addition to HMMs, there are also other movement representations that are interesting in our context, e.g., [5, 9]. However, these approaches share the same problems as the HMM based approaches.

3 Preliminaries of HMMs

A hidden Markov model is a finite state machine extended in a probabilistic manner, and is defined as a triple $\lambda = (A, B, \pi)$. Here, B defines the output distributions $b_i(x) = \mathrm{P}(x|q_t = i)$ of the states. The transition matrix $A = (a_{ij})$ defines the transition probability between

the hidden states $i, j = 1, \ldots, N$, and encodes as such the temporal behavior. The initial state distribution is defined by the vector π.

In our approach continuous left-right HMMs are used with a single Gaussian output distribution $b_i(x) = \mathcal{N}(x|\mu_i, \Sigma_i)$ for each state i. State transitions are either self-transitions or transitions to the successor, i.e., other transition probabilities are zero. In such a model of a single trajectory $X = x_1 \ldots x_t \ldots x_T$ each Gaussian $\mathcal{N}_i(x) := b_i(x)$ "covers" some part of the trajectory, where the state i increases meanwhile the time of the trajectory evolves. In the case of multiple trajectories the Gaussians capture the variance of the training input, but in addition, an HMM compensates for different progression rates of the training trajectories. Obviously, the synthesis of movements is straightforward for this type of HMMs.

For a comprehensive introduction to HMMs, we refer to [3, 8]. The most important algorithms of the HMM framework are mentioned in the following example of a recognition framework. For recognition or classification HMMs are generally used as follows: For each sequence class k an HMM λ^k is trained by maximizing the likelihood function $\mathrm{P}(\mathcal{X}|\lambda)$ with the Baum/Welch expectation maximization (EM) algorithm [8] for a given training data set $\mathcal{X}^k$. The classification of a specific output sequence $X = x_1 \ldots x_T$ is done by identifying with that class k, for which the likelihood $\mathrm{P}(X|\lambda^k)$ is maximal. Here, the forward/backward algorithm [8] is used to efficiently calculate these likelihoods.

One obvious approach for handling whole classes of parameterized actions for the purpose of parameter recognition is a mixture-of-experts approach [4] with sampling of the parameter space. However, this approach suffers from the great number of HMMs needed to be trained and stored for all possible trajectories. Therefore, we introduce the parameterization of the movements as additional model parameters, which also is the basic idea in [10].

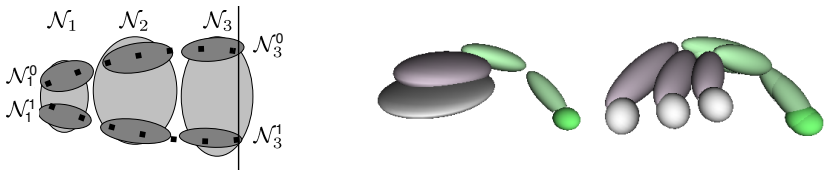


Figure 2: *Left:* The upper three dark ellipsoids are depicting Gaussians $\mathcal{N}_1^0, \ldots, \mathcal{N}_3^0$ of states $i = 1, 2, 3$ of an HMM λ^0 that is trained by sequences, that are beginning on the left, and are leading to the upper of the vertical line. In this case the parameter of sequences is $u = 0$. The dots sketch one of these training sequences. Similarly, the lower three ellipsoids of λ^1 model sequences with parameter $u = 1$. In addition, the Gaussians $\mathcal{N}_i$ of a global model λ are indicated in light gray. In this case, λ is trained with all training sequences. — *Right:* Some Gaussians of the finger tip component of a global HMM trained for our pointing movement are depicted in the middle. The index finger trajectories are leading from the right (green ball) to the left, where the disc like ellipsoid models the finger positions for all pointed at positions at table-top. This global HMM is used to setup the local exemplar HMMs for specific positions in a synchronized way (right).

4 Parametric HMM Framework

The main idea of our approach for handling whole classes of parameterized actions is a supervised learning approach where we generate an HMM for novel action parameters by local linear interpolation of exemplar HMMs that were previously trained on exemplar movements with known parameters. The generation of an HMMs λ^{ϕ} for a specific parameter is carried out by component-wise linear interpolation of the nearby exemplar models. That results, e.g., in the case of a single parameter u and two exemplar models λ^{u}, $u = 0, 1$, in a state-wise generation of the Gaussian $\mathcal{N}_i^u(x) = \mathcal{N}(x|\mu_i^u, \Sigma_i^u)$ for the model λ^{u}, where

$$\mu_i^u = (1-u)\mu_i^0 + u\mu_i^1 \qquad \Sigma_i^u = (1-u)\Sigma_i^0 + u\Sigma_i^1. \tag{1}$$

This situation of two exemplar models λ^{u}, $u = 0, 1$ is sketched in Fig. 2. In the case of such an arrangement, the state-wise interpolation results in a good model λ^{u} for trajectories with parameters $u \in [0, 1]$. But this interpolation approach works *only if* two corresponding states of the two exemplar HMMs model the same semantical part of the trajectory. Therefore, a state-wise alignment is necessary which we describe in Sec. 4.1 below. The expansion to the multi-variate case of parameterization ϕ is straightforward, e.g., by using bilinear ($\phi = (u, v)$) or trilinear interpolation.

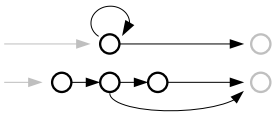


Figure 3: Time duration workaround by replacing each state by several pseudo states.

4.1 Synchronized State Setup for HMMs

As mentioned above, it has to be made sure that corresponding states of exemplar HMMs model the same semantical parts of movements. This task is somehow similar to those handled by standard dynamic time warping (DTW) approaches for aligning two sequences. However, here, an approach for the hidden states is needed, and this, in addition, in the presence of many sequences.

The underlying idea is to set up local exemplar HMMs λ^{ϕ} by using the invariance of HMMs to temporal variations. We proceed in two steps: At first a global HMM λ is trained based on the whole training set $\mathcal{X}$ of movements of different parameterizations ϕ, but of same type. Such a global HMM is sketched in Fig. 2 by the light gray Gaussians. The situation that movements of different parameterizations are covered in such a symmetrical way as in Fig. 2 can be enforced, in some way, by forcing the hidden state sequences to pass the states always in the sequential order from state 1 to N. This is caused by the choice of the type of left-right model, and by allowing only sequences that start in the first and end in the last state. In addition, the invariance of the HMM to temporal variations needs to be constraint (similar to constraining the warping in standard DTW approaches). We addressed this by adding explicit time durations to the states of the HMM [8]. To circumvent the numerical problems of scalings (see, [8]), we replaced each

state of the left-right HMM with some pseudo states that share one Gaussian (compare Fig. 3). This forces the hidden states sequences to stay in a state, e.g., as in Fig. 3, for at least two and for maximal three time steps.

In the second step, consider the partial training set $\mathscr{X}^\phi$ of a specific parameterization ϕ. On this training set we train a local exemplar HMM λ^ϕ while using the global HMM λ for the alignment. In the framework of the EM algorithm, we do this by computing the local model λ^ϕ for $\mathscr{X}^\phi$ by using λ as an initial configuration and by fixing the means of the Gaussians after the first EM iteration. It is worth to note, that this gives the wanted result: In the first E step of the EM algorithm, the posterior probabilities $\gamma_t^k(i) = \mathrm{P}(q_t = i|X^k,\lambda)$ of being in state i at time t given the global model are computed for each sequence $X^k = x_1^k \dots x_T^k$ of the training set $\mathscr{X}^\phi$. Thus, $\gamma_t^k(i)$ defines somehow the "responsibility" of each state i for generating x_t^k. Then, in the M step the mean μ_i is re-estimated for each Gaussian of each state i as a $\gamma_t^k(i)$-weighted mean:

$$\mu_i = \frac{\sum_{t,k} \gamma_t^k(i) x_t^k}{\sum_{t,k} \gamma_t(i)} \tag{2}$$

Now, consider Fig. 2, and the depicted upper sequence $x_1 x_2 \dots x_7$. The responsibilities $\gamma_t(i)$ of state $i = 1$ are only large for the first outputs, e.g., $t = 1,2$, and are very small for $t > 2$ if one considers the position of the Gaussian $\mathscr{N}_1$. Thus, the mean μ_1^0 of $\mathscr{N}_1^0$ as given by Eq. (2), lies between x_1 and x_2, as required.

4.2 Synthesis, Recognition, and Parameters

Consider a grasp position p on a table-top. Then, synthesis is done as follows: At first, four HMMs $\lambda^i{}_{,i=1,\dots,4}$ with closest associated grasp positions p^i are chosen under the constraint that at least three of the p^i are strongly not collinear and that p lies accurately in the convex hull of $\{p^i\}$. Then, the bilinear interpolation parameters u, v are estimated such that the interpolated point p^{uv} approximates p best. Then, the model λ^{uv}, i.e., the sequence $\mu_1^{uv} \dots \mu_N^{uv}$ of the Gaussians, is calculated. Afterwards, this sequence is expanded to a function $f(t)$ by spline interpolation, if needed, with respect to the time durations coded in the transition matrix.

The recognition of the type and the parameterization of the recognized type of a parameterized movement is straightforward compared to the nonparametric case of classification. Consider a given sequence X. We proceed in two steps: First, for each possible movement type k the most likely parameter ϕ^k of the corresponding parameterized HMM λ_k^ϕ is estimated. Therefore, we maximize $l_k(\phi) = P(X|\lambda_k^\phi)$ under the constraint of senseful values $\phi \in [0-\varepsilon, 1+\varepsilon]^d$ by using the gradient descent. Then, the movement is classified as that class k of highest likelihood $l_k(\phi^k)$. In addition, the parameter ϕ^k gives the most likely parameterization. That identifies in the table-top scenario the pointed at position p^{uv}, which is given by the bilinear interpolation parameters $(u, v) = \phi^k$.

In our table-top experiments there are up to nine exemplar HMMs in the PHMM. Therefore, the estimate of the parameter ϕ is done in a hierarchical way (starting with a first estimate ϕ based on the PHMM given by bilinear interpolation of the four outermost exemplar HMMs, and ending with a refinement of ϕ based on the exemplar HMMs nearby the previous estimate).

5 Experiments

In our experiments we focus our considerations on pointing (see Fig. 1) and grasping actions, which are in most human robot interaction scenarios probably the two most important movements. Both are performed in a very similar way, starting and ending in the same base position (arm hanging down). — In this section we provide off-line and on-line evaluations of our PHMMs. First we evaluate the precision of recognition and synthesis for marker data. Then, we test our PHMMs in on-line setups. Concerning online recognition and synthesis, we have results in a form of an online recognition, and a robot motor control demo.

The motion data of our systematic off-line experiments is acquired (Fig. 1) with 60Hz with an eight camera visual marker motion capture system of *Vicon*. The recognition and synthesis experiments are based on seven 3D points located at different segments of a human body. The seven data points are: sternum; shoulder, and elbow of the right arm; index finger, its knuckle, and thumb of the right hand.

The exemplar positions at table-top form a regular raster, which covers a region of 80cm $\times$ 30cm (width $\times$ depth). For training, a 3×3 raster is used, where 10 repetitions have been recorded for each exemplar position and each action type (pointing, grasping). For evaluation, a 5×7 raster is used with 4 repetitions for each position to allow a good evaluation statistic. All in all several hundreds of repetitions for testing.

5.1 Training: Setup of PHMMs

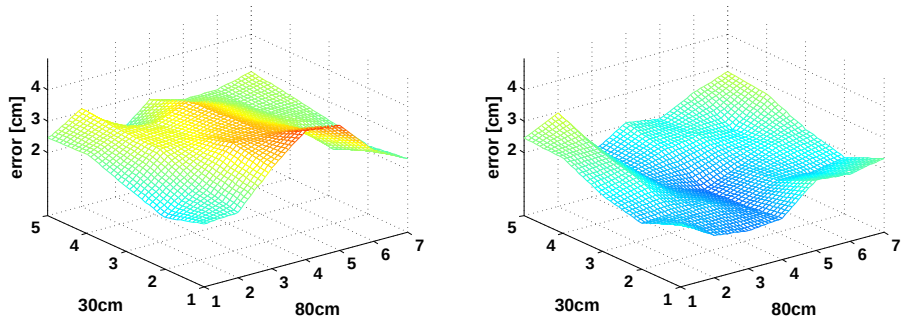
Training and setup of the exemplar PHMMs for grasping and pointing is done as described in Sec. 4.1. We used PHMMs of 20 states, where the hidden state sequences are forced to stay between 4 and 6 steps in each state. The training sequences are rescaled to 100 samples. We train the PHMMs based on data of the full 3×3 raster (9 exemplar HMMs) or based on a 2×2 raster, which consists only of the outer most exemplar positions of the 3×3 raster. These PHMMs will be refered in the following as 3×3 or 2×2 PHMM of grasping or pointing.

5.2 Synthesis

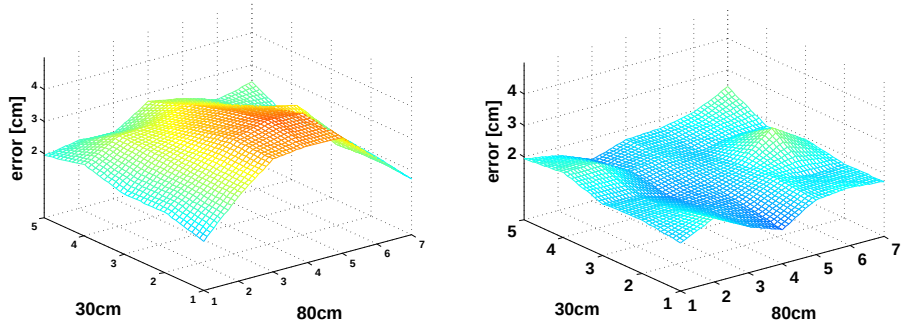
Synthesis is done as described in Sec. 4.2 with the setup described in the section above. The performance of synthesis is systematically evaluated by plotting the synthesis error for each 5×7-raster position. Therefore, the error is calculated as a distance measure between each synthesized movement $f(t)$ and an average $\overline{f}(t)$ of the four test exemplars of the raster position. The averaging is done by using 80 state HMMs. Both movements $f(t)$ and $\overline{f}(t)$ are functions $f(t) = (f_i(i))_{i=1}^{7}$ of stacked 3D trajectories $f_i(t)_{i=1,\ldots,7}$ (elbow, wrist...). The error ε is calculated as the route-mean-square error between the time warped synthesis, $f(t)$, and the average $\overline{f}(t)$ of the test movements:

$$\varepsilon = \sqrt{\frac{1}{7}\sum_{i=1}^{7}\int (f_i(\alpha(t)) - \overline{f}_i(\overline{\alpha}(t)))^2 \mathrm{d}t \Big/ \int \alpha(t)\mathrm{d}t}, \tag{3}$$

where $\alpha(t)$ and $\overline{\alpha}(t)$ are warping functions. As the starting and ending points of the reference $\overline{f}(t)$ do vary slightly, the first and last 10% of the sequences are not considered.



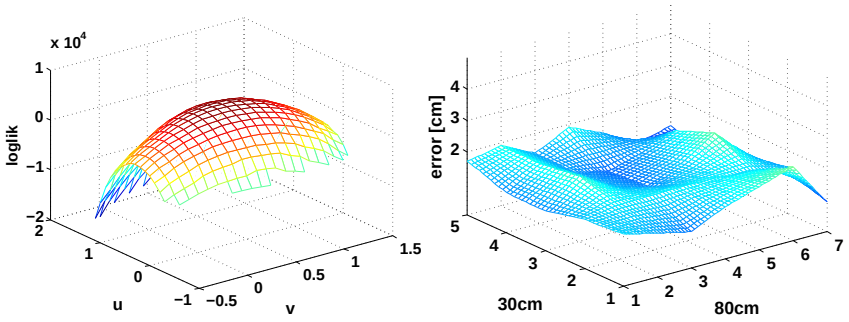
(a) Pointing Synthesis, 2×2 PHMM (b) Pointing Synthesis, 3×3 PHMM



(c) Grasping Synthesis, 3×3 PHMM (d) Grasping Synthesis, 3×3 PHMM

Figure 4: Error of Synthesis at 7×5 raster positions for different PHMMs resolutions.

The Fig. 4 (a), and (b) compare the synthesis errors of 2×2 and 3×3 exemplar PHMMs. Clearly, the performance in the middle of the covered region increases, if the 3×3 PHMM is used. Fig. 4 (c), and (d) show the fact, that the results for the grasping action are very similar to those of the pointing actions. The synthesis errors are approximately 1.8cm for our PHMM for grasping and pointing, if one neglects the outer regions, where the pose of the person is extremely stretched.



(a) Loglik of the Model Parameters (u, v) (b) Recognition of Pointing, 3×3 PHMM

Figure 5: Loglik $\log P(X|\lambda^{uv})$ for a fix sequence (left). Recognition Error Plot (right).

5.3 Recognition

Here, we considered: the performance of recognizing the parameter of a movement, and the rate of correct classifications of movement types, and the robustness to noise.

In advance, it is worth to take a look at Fig. 5 (a), which gives a hint that the maximization of the log likelihood $l(u,v) = \log P(X|\lambda^{uv})$ given a movement is tractable by standard optimization techniques (smoothness, strict concavity). Hence, the most likely parameter (u,v) can be estimated. The error for each position of the 5×7 raster are calculated as the average deviation of the estimated position (u,v) at table-top and the ground truth position for the test movements. The recognition performance of the positions behaves (see Fig. 5) very similar to the results of synthesis. The performance increases similar to the synthesis in the inner region for our 3×3 PHMM compared to the 2×2 PHMM (not depicted). The rate of right-classified types of the 280 grasping and pointing test movements decreases from 94% to 93% by using the 3×3 PHMMs in stead of the 2×2 PHMMs.

We tested the robustness of recognizing the parameterization of movements by adding Gaussian noise to each component of the samples of the movements. Here, we realized no significant influence for independent distributed noise with $\sigma < 15$cm. Obviously, that is caused by the great number of samples of a sequence.

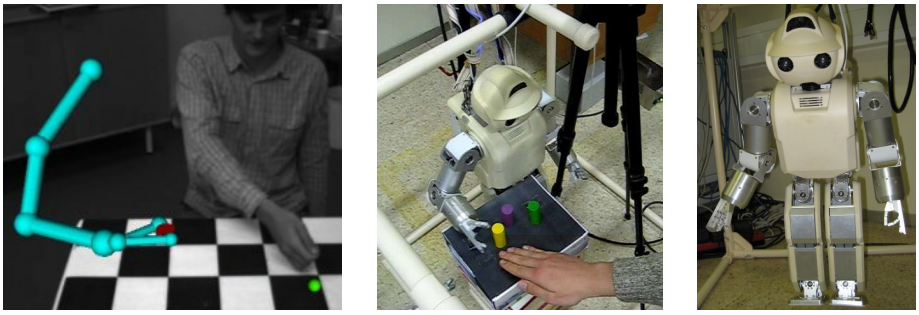


Figure 6: *Left: Online Recognition Demo.* A person is advising a virtual robot arm to relocate objects (currently, a red one is grasped by robot). The ball nearby the person's hand indicates the recognized position, and a high likelihood (green) of pointing. — *Middle: Robot Synthesis Demo.* A person advises the robot HOAP-3 (right), which object to clean up next.

5.4 Online Recognition and Synthesis

In addition to our off-line experiments, we have also implemented and tested our approach for online and real-time processing. We have done two implementations: In our first implementation [file1.avi], we have used an augmented reality setup, see Fig. 6 (left), with an animated robot arm, a stereo camera rig, and two objects on a table. The aim of the demo is to let the human first point at the object to be grasped by the virtual robot arm and then to point at the position on the table where the robot arm should place the object. The stereo camera rig was appropriately calibrated, and the PHMMs were used to identify what action the human performed. In case that a pointing action was observed, the parameters were extracted in order to identify at which object the human had pointed. For tracking the body parts, we used our 3D body tracker.

In our second implementation [file2.avi] we replaced the animated robot arm with the humanoid robot HOAP-3 by Fujitsu Fig. 6 (right). Starting point of that demo was a children's toy box with special holes for special object shapes. The aim was to tell the robot though pointing and grasping gestures which object belongs into which hole such that the robot would be able to learn and perform the appropriate actions later in a different setup. For training and testing the objects could be anywhere on the table. Again, we used our PHMMs to identify the teacher's actions and to synthesize movements for HOAP-3.

6 Conclusion

We have presented and evaluated a novel approach to handle recognition and synthesis of parametric movements (movements of particular type, or sematic). The basic idea is to incorporate the parameterization of the movements into the HMM (PHMM). Contrary to Wilson and Bobick [10], we deal with full arm movements (stacked trajectories), the recognition of the parameters of movements, likewise its type, and synthesis of movements. Instabilities in the training process are circumvent by restricting the dynamic time warping capabilities of HMMs. The experiments show the applicability of our approach for synthesis and recognition of movements (errors $\approx$2cm). The classification rate is $\approx 94\%$, without any kind of diagnostic features, for very similar movements.

Acknowledgment. This work was partially funded by PACO-PLUS (IST-FP6-IP-027657).

References

[1] T. Asfour, K. Welke, A. Ude, P. Azad, J. Hoeft, and R. Dillmann. Perceiving objects and movemetns to generate actions on a humanoid robot. In *Proc. Workshop: From features to actions – Unifying perspectives in compnational and robot vision, ICRA*, Rome, April 2007.

[2] Gutemberg Guerra-Filho and Yiannis Aloimonos. A sensory-motor language for human activity understanding. *HUMANOIDS, 2006*.

[3] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[4] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[5] C. Lu and N. Ferrier. Repetitive Motion Analysis: Segmentation and Event Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):258–263, 2004.

[6] T. Moeslund, A. Hilton, and V. Krueger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–127, 2006.

[7] D. Ormoneit, H. Sidenbladh, M.J. Black, and T. Hastie. Learning and Tracking Cyclic Human Motion. In *Workshop on Human Modeling, Analysis and Synthesis at CVPR*, Hilton Head Island, South Carolina, June 13-15 2000.

[8] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.

[9] D.D. Vecchio, R.M. Murray, and P. Perona. Decomposition of Human Motion into Dynamics-based Primitives with Application to Drawing Tasks. *Automatica*, 39(12):2085–2098, 2003.

[10] Andrew D. Wilson and Aaron F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on PAMI*, 21(9):884–900, 1999.

Visual Recognition of Grasps for Human-to-Robot Mapping

Hedvig Kjellström Javier Romero Danica Kragić
Computational Vision and Active Perception Lab
Centre for Autonomous Systems
School of Computer Science and Communication
KTH, SE-100 44 Stockholm, Sweden
hedvig,jrgn,dani@kth.se

***Abstract*— This paper presents a vision based method for grasp classification. It is developed as part of a Programming by Demonstration (PbD) system for which recognition of objects and pick-and-place actions represent basic building blocks for task learning. In contrary to earlier approaches, no articulated 3D reconstruction of the hand over time is taking place. The indata consists of a single image of the human hand. A 2D representation of the hand shape, based on gradient orientation histograms, is extracted from the image. The hand shape is then classified as one of six grasps by finding similar hand shapes in a large database of grasp images. The database search is performed using Locality Sensitive Hashing (LSH), an approximate k-nearest neighbor approach. The nearest neighbors also give an estimated hand orientation with respect to the camera. The six human grasps are mapped to three Barret hand grasps. Depending on the type of robot grasp, a precomputed grasp strategy is selected. The strategy is further parameterized by the orientation of the hand relative to the object. To evaluate the potential for the method to be part of a robust vision system, experiments were performed, comparing classification results to a baseline of human classification performance. The experiments showed the LSH recognition performance to be comparable to human performance.**

I. Introduction

Programming service robots for new tasks puts significant requirements on the programming interface and the user. It has been argued that the Programming by Demonstration (PbD) paradigm offers a great opportunity to unexperienced users for integrating complex tasks in the robotic system [1]. The aim of a PbD system is to use natural ways of human-robot interaction where the robots can be programmed for new tasks by simply observing human performing the task. However, representing, detecting and understanding human activities has been proven difficult and has been investigated closely during the past several years in the field of robotics [2], [3], [4], [5], [6], [7], [8].

In our work, we have been studying different types of object manipulation tasks where grasp recognition represents one of the major building blocks of the system [1]. Grasp recognition was performed using magnetic trackers [7], together with data gloves the most common way of obtaining the measurements in the robotics field. Although magnetic trackers and datagloves deliver exact values of hand joints, it is desirable from a usability point of view that the user demonstrates tasks to the robot as naturally as possible; the use of gloves or other types of sensors may prevent a natural grasp. This motivates the use of systems with visual input.

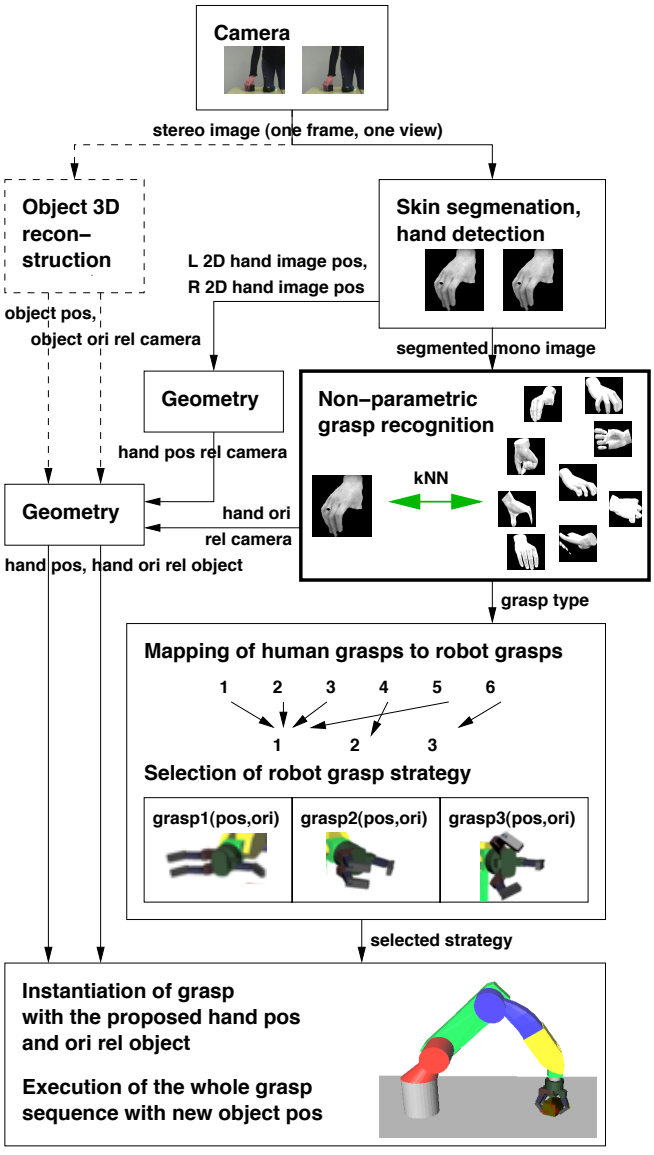
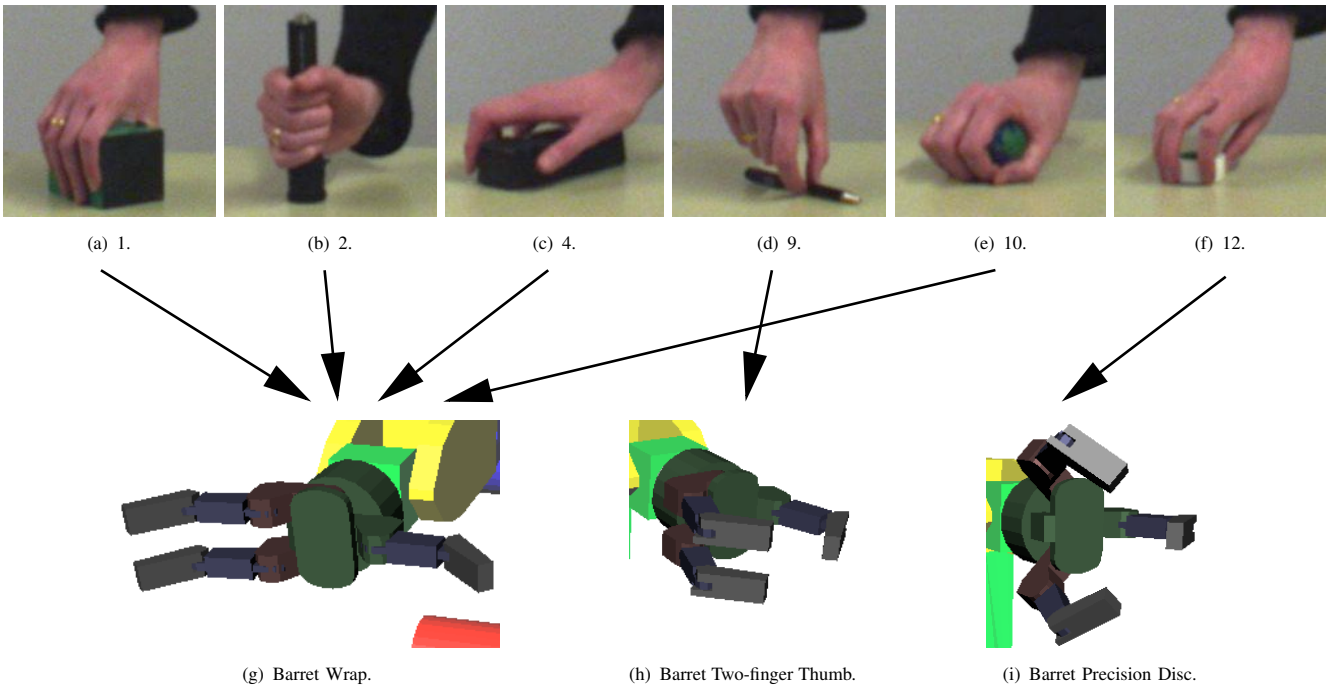


Fig. 1. Human grasps are recognized and mapped to a robot. From *one time frame* of video, the hand is localized and segmented from the background. The hand orientation relative to the camera, and type of grasp is recognized by nearest neighbor comparison of the hand view with a database, consisting of synthesized views of all grasp types from different orientations. The human grasp class is mapped to a corresponding robot grasp, and a predefined grasp strategy, *the whole approach-grasp-retreat sequence*, for that grasp is selected. The strategy is parameterized with the orientation and position of the hand relative to the object, obtained from the hand and object positions and orientations relative to the camera. (In our experiments, the object position and orientation were obtained by hand.)



(a) 1. (b) 2. (c) 4. (d) 9. (e) 10. (f) 12.

(g) Barret Wrap. (h) Barret Two-finger Thumb. (i) Barret Precision Disc.

Fig. 2. The six grasps (numbered according to Cutkosky's grasp taxonomy [9]) considered in the classification, and the three grasps for a Barret hand, with human-robot class mappings ((a,b,c,e)→(g), (d)→(h), (f)→(i)) shown. a) Large Diameter grasp, 1. b) Small Diameter grasp, 2. c) Abducted Thumb grasp, 4. d) Pinch grasp, 9. e) Power Sphere grasp, 10. f) Precision Disc grasp, 12. g) Barret Wrap. h) Barret Two-finger Thumb, i) Barret Precision Disc.

Vision based recognition of a grasping hand is a difficult problem, due to the self occlusion of the fingers as well as the occlusion of the hand by the grasped object [10], [11], [12], [13]. To simplify the problem, some approaches use optical markers [14], but markers make the system less usable when service robot applications are considered. We therefore strive to develop a markerless grasp recognition approach.

Figure 1 outlines the whole mapping procedure. Although the scientific focus of this paper is on the classification on human grasps, the classification method should be thought of as part of the whole mapping procedure, which consists of three main parts: The human grasp classification, the extraction of hand position relative to the grasped object (with object detection not implemented for our experiments), and the compilation of a robot grasp strategy, parameterized by the type of grasp and relative hand-object orientation and position, described in Section VI.

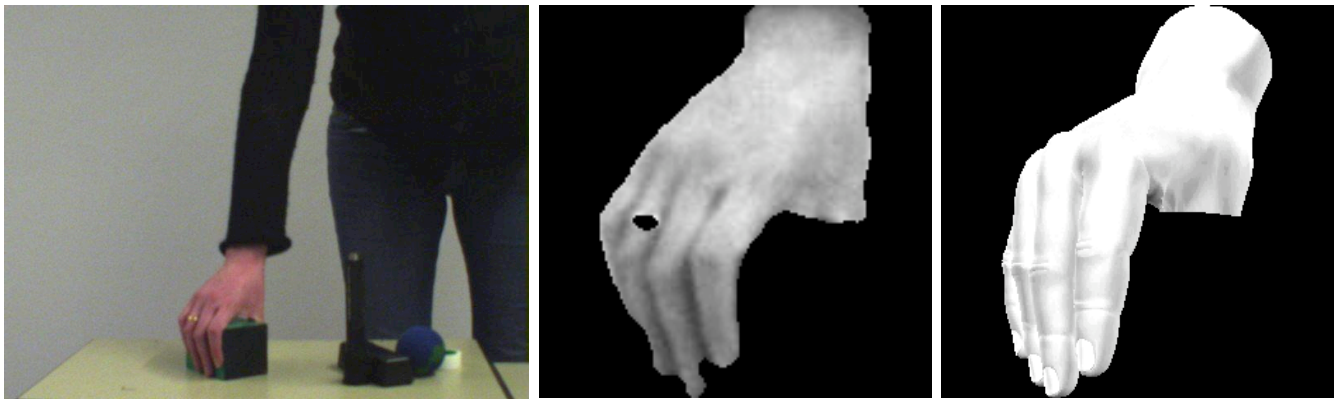
The main contribution of this paper is a non-parametric method for grasp recognition. While articulate 3D reconstruction of the hand is straightforward when using magnetic data or markers, 3D reconstruction of an unmarked hand from images is an extremely difficult problem due to the large occlusion [10], [11], [12], [13], actually more difficult than the grasp recognition problem itself as discussed in Section II. Our method can classify grasps and find their orientation, from a single image, from any viewpoint, without building an explicit representation of the hand, similarly to [12], [15]. Other grasp recognition methods (Section II) consider only a single viewpoint or employ an invasive sensing device such as datagloves, optical markers for motion capture, or magnetic sensors.

The general idea to recognize the human grasp and select a precomputed grasping strategy is a secondary contribution of the paper, since it differs from the traditional way to go about the mapping problem [7]; to recover the whole 3D pose of the human hand, track it through the grasp, and then map the motion to the robot arm. A recognition-based approach such as ours avoid the difficult 3D reconstruction problem, and is also much more computationally efficient since it only requires processing of a single video frame.

The grasp recognition problem is here formalized as the problem of classifying a hand shape as one of six grasp classes, labeled according to Cutkosky's grasp taxonomy [9]. The classes are, as shown in Figure 2a-f, Large Diameter grasp, Small Diameter grasp, Abducted Thumb grasp, Pinch grasp, Power Sphere grasp and Precision Disc grasp.

The input to the grasp classification method is a single image (one time instance, one camera view point) from the robot's camera. The hand is segmented out using skin color segmentation, presented in more detail in Section III. From the segmented image, a representation of the 2D hand shape based on gradient orientation histograms is computed as presented in Section IV. A large set of synthetic hand views from many different viewpoints, performing all six types of grasps has been generated. Details are given in Section III. The new hand shape is classified as one of the six shapes by approximate k-nearest neighbor comparison using Locality Sensitive Hashing (LSH) [16]. Along with the grasp class, the estimated orientation of the hand relative to the camera is obtained by interpolating between the orientations of the found nearest neighbors. This is presented in Section V.

Experiments presented in Section VII show the method to



(a) Input image $\mathbf{I}$. (b) Hand image $\mathbf{H}$. (c) Synthetic hand image $\mathbf{H}^{\mathrm{synth}}$.

Fig. 3. Processing of image data. a) Input $\mathbf{I}$ from the robot, grabbed with an AVT Marlin F-080C camera. b) Segmented hand image $\mathbf{H}$. c) Synthetic view of hand $\mathbf{H}^{\mathrm{synth}}$, generated in Poser 7.

perform comparably to humans, which indicates that it is fit to be included into complex vision system, such as the one required in a PbD framework.

II. Related Work

Classification of hand pose is most often used for gesture recognition, e.g. sign language recognition [12], [17]. These applications are often characterized by low or no occlusion of the hands from other objects, and a well defined and visually disparate set of hand poses; in the sign language case they are designed to be easily separable to simplify fast communication. Our problem of grasp recognition differs from this application in two ways. Firstly, the grasped object is usually occluding large parts of the grasping hand. We address this by including expected occlusion in our dataset; occluding objects are present in all example views (Section III). Secondly, the different grasping poses are in some cases very similar, and there is also a large intra-class variation, which makes the classification problem more difficult.

Related approaches to grasp recognition [14], [18] first reconstruct the hand in 3D, from infrared images [18] or from an optical motion capture system which gives 3D marker positions [14]. Features from the 3D pose are then used for classification. The work of Ogawara et al. [18] views the grasp recognition problem as a problem of shape reconstruction. This makes their results hard to compare to ours. In addition, they also use a wide baseline stereo system with infrared cameras, which makes their approach difficult to adopt in a case of a humanoid platform.

The more recent work of Chang et al. [14] learns a non-redundant representation of pose from all 3D marker positions – a subset of features – using linear regression and supervised selection combined. In contrast, we use a completely non-parametric approach where the classification problem is transformed into a problem of fast LSH nearest neighbor search (Section IV). While a linear approach is sufficient in the 3D marker space of Chang et al. [14] , the classes in the orientation histogram space are less Gaussian shaped and more intertwined, which necessitates a non-linear or non-parametric classifier as ours.

Using 3D motion capture data as input, Chang et al. [14] reached an astonishing recognition rate of up to 91.5%. For the future application of teaching of service robots it is however not realistic to expect that the teacher will be able or willing to wear markers to provide the suitable input for the recognition system. 3D reconstructions, although with lower accuracy, can also be achieved from unmarked video [19], [20]. However, Chang et al. [14] note that the full 3D reconstruction is not needed to recognize grasp type. Grasp recognition from images is thus an easier problem than 3D hand pose reconstruction from images, since fewer parameters need to be extracted from the input. We conclude that the full 3D reconstruction is an unnecessary (and error prone) step in the chain from video input to grasp type.

Our previous work [7] considered an HMM framework for recognition of grasping sequences using magnetic trackers. Here, we are interested in evaluating a method that can perform grasp classification based on a single image only, but it should be noted that the method can easily be extended for use in a temporal framework.

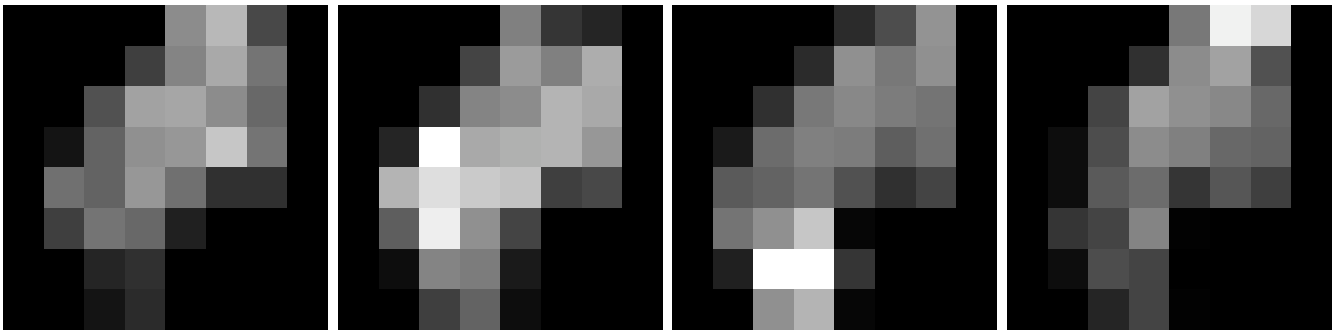
III. Extracting the Hand Image

Since the robot grasp strategies are predefined, and only parameterized by the hand orientation, position and type of grasp, there is no need for the human to show the whole grasp procedure; only one time instance is enough (for example, the image that is grabbed when the human tells the robot ”now I am grasping”).

The input to the recognition method is thus a single monocular image $\mathbf{I}$ from the a camera mounted on the robot. For our experiments, we use an AVT Marlin F-080C camera. An example of an input image is shown in Figure 3a. Before fed into the recognition, the image is preprocessed in that the grasping hand is segmented from the background.

A. Segmentation of hand images

The hand segmentation could be done using a number of modalities such as depth (estimated using stereo or an active sensor) or color. We choose to use skin color segmentation;



(a) $\frac{\pi}{8}$. (b) $\frac{3\pi}{8}$. (c) $\frac{5\pi}{8}$. (d) $\frac{7\pi}{8}$.

Fig. 4. Gradient orientation histograms from the hand image $\mathbf{H}$ of Figure 3b, with $B = 4$ bins, on level $l = 1$ in the pyramid of $L = 4$ levels (spatial resolution 8×8). a) Bin 1, orientation $\frac{\pi}{8}$. b) Bin 2, orientation $\frac{3\pi}{8}$. c) Bin 3, orientation $\frac{5\pi}{8}$. d) Bin 4, orientation $\frac{7\pi}{8}$.

the details of the method used are described in [21]. To remove segmentation noise at the borders between background and foreground, the segmentation mask is median filtered three times with a 3×3 window.

The segmented image $\hat{\mathbf{H}}$ is cropped around the hand and converted from RGB to grayscale. An example of the resulting hand image $\mathbf{H}$ is shown in Figure 3b.

B. Generation of synthetic hand images for the classification

The fact that the classification method (Section V) is non-parametric and that no explicit model of the hand is built (Section IV) means that a very large set of examples, from many different views, is needed for each grasp.

As it is virtually intractable to generate such training sets using real images, we use a commercial software, Poser 7, to generate synthetic views $\mathbf{H}^{\text{synth}}$ of different hand configurations. Poser 7 supplies a realistic 3D hand model which can be configured by bending the finger joints. For our purposes, the model was configured by hand into the 6 iconic grasps, which were a little exaggerated to provide clear distinctions between the classes. 900 views of each configuration were generated, with viewing angles covering a half-sphere in steps of 6 degrees in camera elevation and azimuth; these are the views which can be expected by a robot with cameras above human waist-height. The synthetic hand was grasping an object, whose shape was selected to be typical of that grasp [9]. The object was black (as the background), and occluded parts of the hand as it would in the corresponding real view of that grasp. This will make the synthetic views as similar as possible to the real views (e.g. Figure 3b), complete with expected occlusion for that view and grasp. Figure 3c shows such a database example.

The synthetic images $\mathbf{H}^{\text{synth}}$ can be seen as ideal versions of the segmented and filtered real hand images $\mathbf{H}$. Note that the recognition method is tested (Section VII) using real hand images prepared as described in the previous subsection, and that the synthetic images are used only for the database. Note further that the hand in the database is not the same as the hand in the test images.

IV. Image Representation

For classification of grasps, we seek a representation of hand views (Figures 3b and 3c) with as low intra-class variance, and as high inter-class variance as possible. We choose gradient orientation histograms, frequently used for representation of human shape [22], [23].

Gradient orientation $\mathbf{\Phi} \in [0, \pi)$ is computed from the segmented hand image $\mathbf{H}$ as

$$\mathbf{\Phi} = \arctan(\frac{\partial \mathbf{H}}{\partial y} / \frac{\partial \mathbf{H}}{\partial x}) \tag{1}$$

where x denotes downward (vertical) direction and y rightward (horizontal) direction in the image.

From $\mathbf{\Phi}$, a pyramid with L levels of histograms with different spatial resolutions are created; on each level l, the gradient orientation image is divided into $2^{L-l} \times 2^{L-l}$ equal partitions. A histogram with B bins are computed from each partition. An example of histograms at the lowest levels of the pyramid can be seen in Figure 4.

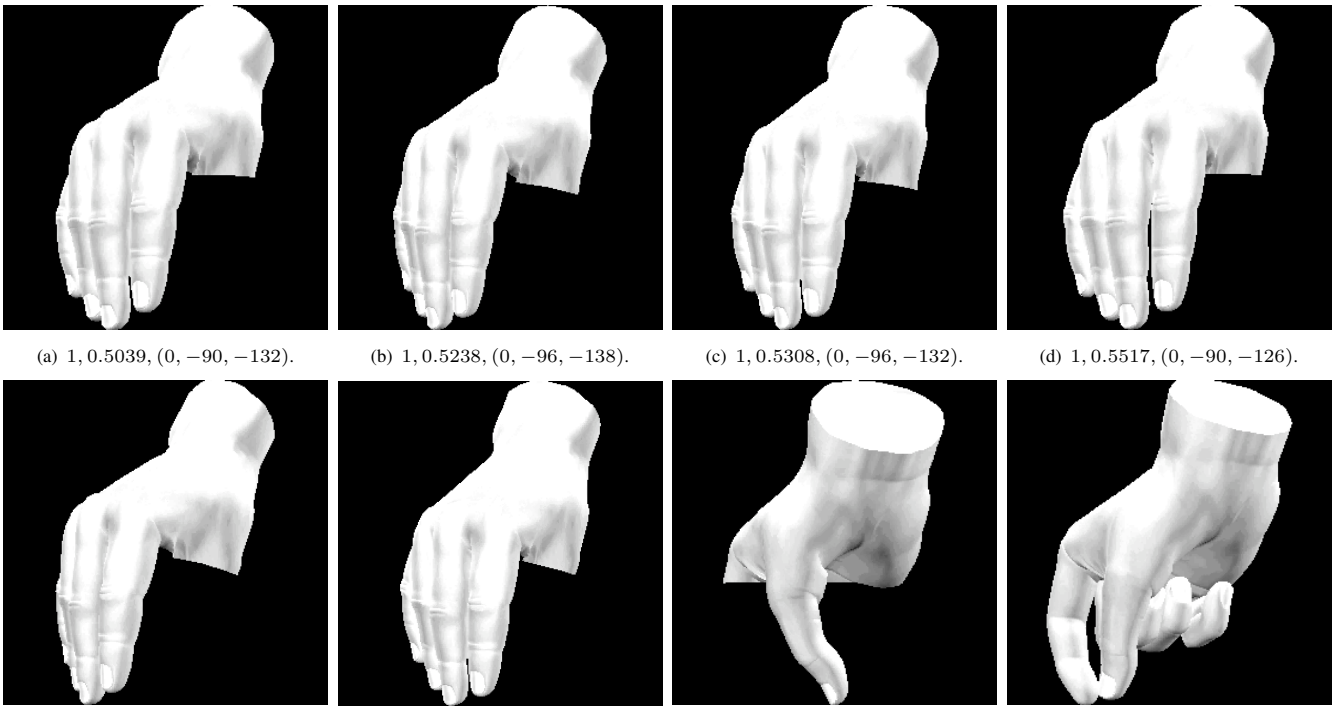
The hand view is represented by $\mathbf{x}$ which is the concatenation of all histograms at all levels in the pyramid. The length of $\mathbf{x}$ is thus $B \sum_{l=1}^{L} 2^{2(L-l)}$. The performance of the classifier is quite insensitive to choices of $B \in [3, 8]$ and $L \in [2, 5]$; in our experiments in Section VII we use $B = 4$ and $L = 3$.

V. Approximate Nearest Neighbor Classification

A database of grasp examples is created by synthesizing $N = 900$ views $\mathbf{H}_{i,j}^{\text{synth}}$ with $i \in [1, M]$, $j \in [1, N]$, from each of the $M = 6$ grasp classes (Section III), and generating gradient orientation histograms $\mathbf{x}_{i,j}$ from the synthetic views (Section IV). Each sample has associated with it a class label $y_{i,j} = i$ and a hand-vs-camera orientation $\mathbf{o}_{i,j} = [\phi_j, \theta_j, \psi_j]$, i.e. the Euler angles from the camera coordinate system to a hand-centered coordinate system.

To find the grasp class $\hat{y}$ and orientation $\hat{\mathbf{o}}$ of an unknown grasp view $\mathbf{x}$ acquired by the robot, a distance-weighted k-nearest neighbor (kNN) classification/regression procedure is used. First, X_k, the set of k nearest neighbors to $\mathbf{x}$ in terms of Euclidean distance $d_{i,j} = \|\mathbf{x} - \mathbf{x}_{i,j}\|$ are retrieved.

As an exact kNN search would put serious limitations on the size of the database, an approximate kNN search method,



(a) $1, 0.5039, (0, -90, -132)$. (b) $1, 0.5238, (0, -96, -138)$. (c) $1, 0.5308, (0, -96, -132)$. (d) $1, 0.5517, (0, -90, -126)$.

(e) $1, 0.5523, (0, -96, -144)$. (f) $1, 0.5584, (0, -102, -132)$. (g) $1, 0.5870, (0, -90, +108)$. (h) $4, 0.6068, (0, -90, +120)$.

Fig. 5. Distance-weighted nearest neighbor classification. a-h) Some of the approximate nearest neighbors to the hand view in Figures 3b, with associated grasp class $y_{i,j}$, distance in state-space $d_{i,j}$, and 3D orientation $\mathbf{o}_{i,j}$.

Locality Sensitive Hashing (LSH) [16], is employed. LSH is a method for efficient ϵ-nearest neighbor (ϵNN) search, i.e. the problem of finding a neighbor $\mathbf{x}_{\epsilon\text{NN}}$ for a query $\mathbf{x}$ such that

$$\|\mathbf{x} - \mathbf{x}_{\epsilon\text{NN}}\| \leq (1+\epsilon)\|\mathbf{x} - \mathbf{x}_{\text{NN}}\| \tag{2}$$

where $\mathbf{x}_{\text{NN}}$ is the true nearest neighbor of $\mathbf{x}$. This is done as (see [16] for details): 1) T different hash tables are created independently. 2) For $t = 1, \ldots, T$, the part of state space in which the dataset $\{\mathbf{x}_{i,j}\}_{i\in[1,M],j\in[1,N]}$ resides is randomly partitioned by K hyperplanes. 3) Every point $\mathbf{x}_{i,j}$ can thereby be described by a K bit binary number $\mathbf{f}_{t,i,j}$ defined by its position relative to the hyperplanes of table t. 4) As the total number of possible values of $\mathbf{f}_{t,i,j}$ is large, a hash function $h(\mathbf{f}_{t,i.j})$ gives the index to a hash table of fixed size H.

The ϵNN distance to the unknown grasp view $\mathbf{x}$ is now found as: 1) For each of the T hash tables, compute hash indices $h(\mathbf{f}_t)$ for $\mathbf{x}$. 2) Let $X_\cup = \{\mathbf{x}_m\}_{m\in[1,N_\cup]}$ be the union set of found examples in the T buckets. The ϵNN distance $\|\mathbf{x} - \mathbf{x}_{\epsilon\text{NN}}\| = \min_{m\in[1,N_\cup]} \|\mathbf{x} - \mathbf{x}_m\|$. In analog, the $\min(N_\cup, k)$ ϵ-nearest neighbors X_k are found as the $\min(N_\cup, k)$ nearest neighbors in $X_\cup$.

The parameters K and T for a certain value of ϵ is dataset dependent, but is learned from the normal data itself [24]. We use $\epsilon = 0.05$.

The computational complexity of retrieval of the ϵNN with LSH [16] is $\mathcal{O}(DN^{\frac{1}{1+\epsilon}})$ which gives sublinear performance for any $\epsilon > 0$. For examples of ϵ-nearest neighbors to the hand in Figure 3b, see Figure 5.

From X_k the estimated class of $\mathbf{x}$ is found as,

$$\hat{y} = \arg\max_i \sum_{j:\mathbf{x}_{i,j}\in X_k} \exp(-\frac{d_{i,j}^2}{2\sigma^2}) \ , \tag{3}$$

i.e. a distance-weighted selection of the most common class label among the k nearest neighbors, and the estimated orientation as

$$\hat{\mathbf{o}} = \frac{\sum_{j:\mathbf{x}_{\hat{y},j}\in X_k} \mathbf{o}_{\hat{y},j} \exp(-\frac{d_{\hat{y},j}^2}{2\sigma^2})}{\sum_{j:\mathbf{x}_{\hat{y},j}\in X_k} \exp(-\frac{d_{\hat{y},j}^2}{2\sigma^2})} \ , \tag{4}$$

i.e. a distance-weighted mean of the orientations of those samples among the k nearest neighbors for which $y_{i,j} = \hat{y}$. (The cyclic properties of the angles is also taken into account in the computation of the mean.) As we can see in Figure 5h, the orientation of a sample from a different class has very low correlation with the real orientation, simply because the hand in a different grasp has a different shape. Therefore, only estimates with the same class label as $\hat{y}$ are used in the orientation regression. All in all, the dependency between the state-space and the global Euler angle space is highly complex, and that is why it is modeled non-parametrically.

The standard deviation σ is computed from the data as

$$\sigma = \frac{1}{\sqrt{2}MN} \sum_i \sum_{j_1,j_2\in[1,N],j_1\neq j_2} \|\mathbf{x}_{i,j_1} - \mathbf{x}_{i,j_2}\| \ , \tag{5}$$

the mean intra-class, inter-point distance in the orientation histogram space [25].

The obviously erroneous neighbors in Figures 5g and 5h could maybe have been avoided with a larger

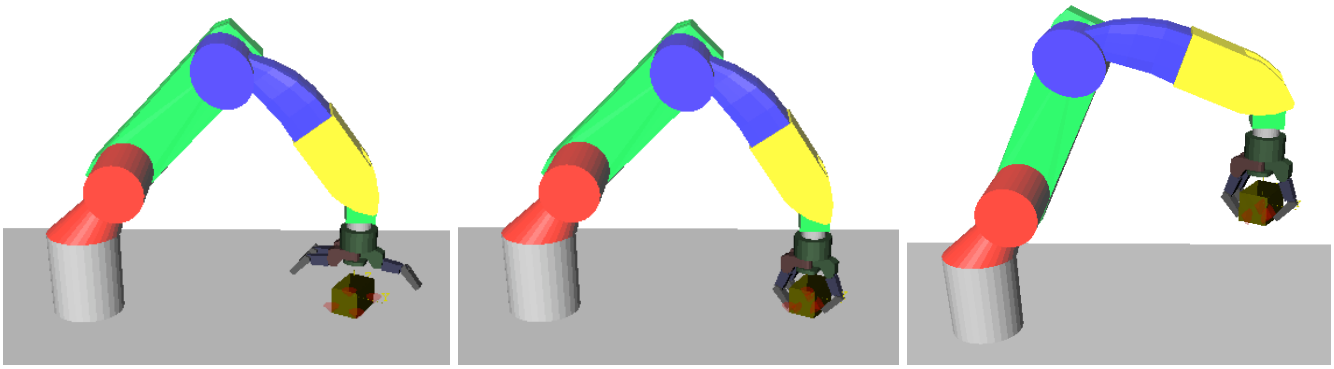


Fig. 6. Barret Wrap grasp, carried out on the same type and size of object as the human Large Diameter grasp shown in Figure 3b.

database containing hands of varying basic shape, such as male/female/skinny/fat/long-fingered/short-fingered hands. The hand in the test images (Figure 3b) is considerably different from the synthetic Poser 7 hand (Figures 3c, 5), and thus their 3D shapes are different even though they take the same pose. This poses no problem to the method in general; since the approximate kNN classification/regression has a sub-linear complexity, the database can be increased considerably to a limited computational cost.

VI. Example-Based Mapping of Grasp to Robot

To illustrate how the grasp classification can be employed for human-to-robot mapping in a pick-and-place scenario, a simulated robot arm is controlled with parameterized predefined grasping strategies as illustrated in Figure 1.

A human-to-robot grasp mapping scheme is defined depending on the type of robot hand used; here we use a Barret hand with three types of grasps as shown in Figure 2. The type of robot grasp defines the preshape of the robot hand.

The hand orientation estimate $\hat{\mathbf{o}}$ relative to the camera, along with the hand position estimate and the estimated position and orientation of the grasped object relative to the camera, are used to derive the estimated position and orientation of the human hand relative to the object, as depicted in Figure 1. The estimation of object position and orientation is assumed perfect; this part of the system is not implemented, instead the ground truth is given in the simulations.

In contrary to related grasp approaches [26], the robot here does not explore a range of approach vectors, but instead directly imitates the human approach vector, encoded in the hand position and orientation relative to the object. This leads to a much shorter computational time at the expense of the non-optimality of the grasp in terms of grasp quality. However, since the the selection of robotic preshape has been guided, the stability of the robotic grasp will be similar to the human one, leading to a non-optimal but successful grasp provided that the errors in the orientation and position estimate are sufficiently small.

An analysis of the robustness to position errors can be found in [26]. For an optimally chosen preshape, there is a error window ≥ 4 cm $\times$ 4 cm about the position of the object, within which the grasps are successful. The positioning of the robot hand can also be improved by fusing the estimated human hand position with an automatic selection of grasping point based on object shape recognition [27].

The robustness to orientation errors depends greatly on the type of grasp and object shape. We investigate the robustness of the Barret Wrap grasp with an approach vector perpendicular to the table (Figure 6). We get good results for orientation errors around the vertical axis of up to 15 degrees. As a comparison, the mean regression error of this orientation (Section VII-B) is on the same order as the error window size, 10.5 degrees, which indicates that the orientation estimation from the grasp classifier should be used as an initial value for a corrective movement procedure using e.g. the force sensors on the hand.

VII. Experimental Results

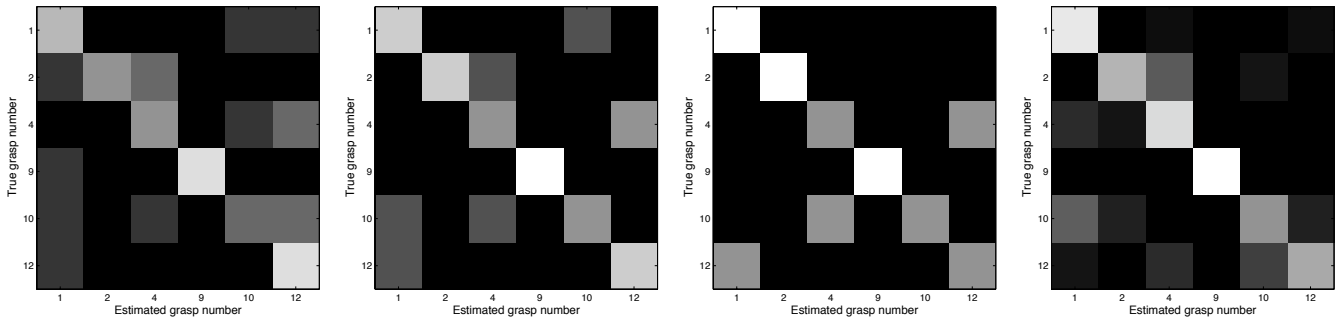
Quantitative evaluations of the grasp classification and orientation estimation performance were made.

For each of the six grasp types, two video sequences of the hand were captured, from two different viewpoints. From each video, three snapshots were taken, one where the hand was starting to reach for the object, one where the hand was about to grasp and one where the grasp was completed. This test set is denoted X.

The test examples from the beginning of the sequences are naturally more difficult than the others, since the hand configuration in those cases are closer to a neutral configuration, thus more alike than the examples taken closer to the completed grasp. It is interesting to study the classification rate for the different levels of neutrality, since it indicates the robustness to temporal errors when the robot grabs the image upon which the classification is based (Section III). In some tests below, we therefore removed the 12 most neutral examples from the test set, denoted X'. In other tests, we kept only the 12 most specific examples, denoted X''.

A. Classification of human grasps: Comparison of LSH and human classification performance

Figures 7a, 7b, and 7c show the confusion matrices for LSH classification of test set X, X', and X'', respectively. Apart from the fact that the performances on X' and X''



(a) LSH, all hand images: 61%. (b) LSH, hand close to object: 71%. (c) LSH, hand grasping object: 75%. (d) Human, all hand images: 74%.

Fig. 7. Confusion matrices for classification of the six grasps. White represents a 100% recognition rate, while black represents 0%. a) LSH performance, all hand images (X): 61% correct classifications. b) LSH performance, images with hand close to object (X'): 71% correct classifications. c) LSH performance, images with hand grasping object (X''): 75% correct classifications. d) Human performance, all hand images (X): 74% correct classifications.

are better than for X, it can be noted that the performance on Pinch grasp (9) and Precision Disc grasp (12) are very good. This is expected since these grasps are visibly very different from the others. Interestingly, it also concords with the mapping to the Barret grasps (Figure 2) in which these grasps have unique mappings while the others all are mapped to the same grasp. Note however that the human grasps map differently to more articulated robot hands.

The error rates alone say little about how the method would perform in a PbD system. The grasp recognition would there interact with methods for object, shape and action recognition, and a perfect performance in an isolated grasp recognition task is probably not needed.

How do we then know what error rate is "enough"? Humans are very good at learning new tasks by visual observation, and reach a near perfect performance on combined object, shape, action and object recognition. Human recognition performance on the same task as our classifier, with the same indata, would thus be a good baseline.

As an important side note, two things can be noted about this comparison. Firstly, in a natural learning situation, a human would use information about the grasped object and the motion of the hand as well. This information is removed for this experiment. As discussed in the Conclusions, we intend to integrate automatic grasp, object and action recognition in the future. Secondly, it is debated how important depth perception is for human recognition; humans perceive depth both through stereo and through prior knowledge about the hand proportions. For this experiment, we disregard depth as a cue in the human experiment.

Figure 7d shows the classification performance of a human familiar with the Cutkosky grasp taxonomy. The human was shown the segmented hand images $\mathbf{H}$ in the set X in random order and was asked to determine which of the six grasp classes they belonged to.

Interestingly, the human made the same type of mistakes as the LSH classifier, although to a lower extent. He sometimes misclassified Power grasp (10) as Large Diameter grasp (1), and Small Diameter grasp (2) as Abducted Thumb grasp (4). This indicates that these types of confusions are intrinsic to the problem rather than dependent on the LSH and training set. Since humans are successful with grasp recognition in a real world setting, these confusions are compensated for in some other way, probably by recognition of shape of the grasped objects. It can also be noted that the human was better at recognizing the most neutral grasps present in X but not in X' or X''.

Overall, the LSH performance is at par with, or slightly worse than human performance. This must be regarded as a successful experimental result, and indicates that the grasp recognition method can be a part of a PbD system with low error rate.

B. Classification of human grasps: Orientation accuracy

Figure 8 shows the mean orientation error for regression with X. The angular displacement of the two coordinate systems corresponds to how far off a robot hand would be in grasping an object without corrective movements during grasping. As noted in Section VI, the orientation estimate from this method should only be regarded as an initial value, from where a stable grasp is found using a corrective movement procedure.

VIII. Conclusions

PbD frameworks are considered as an important area for future robot development where the robots are supposed to learn new task through observation and imitation. Manipulating and grasping known and unknown objects represents a significant challenge both in terms of modeling the observation process and then executing it on the robot.

In this paper, a method for classification of grasps, based on a single image input, was presented. A grasping hand was represented as a gradient orientation histogram; a 2D image-based representation. A new hand image could be classified as one of six grasps by a kNN search among large set of synthetically generated hand images.

On the isolated task of grasp recognition, the method performed comparably to a human. This indicates that the method is fit for use in a PbD system, where it is used in interaction with classifiers of object shape and human actions. The dataset contained grasps from all expected viewpoints and with expected occlusion. This made the

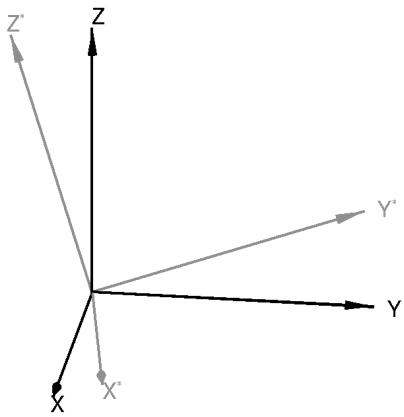


Fig. 8. Mean orientation error, all hand images (X): $(0, 0.29, 0.18)$ radians $= (0, 16.8, 10.5)$ degrees.

method view-independent although no 3D representation of the hand was computed.

The method was considered part of a grasp mapping framework, in which precomputed grasp strategies were compiled based on the detected type of grasp and hand-object orientation.

A. Future Work

It would be interesting to add an object orientation estimation technique to the system, and to execute the grasps on a real robot arm. Furthermore, we will investigate the inclusion of automatic positioning methods into the grasp strategies, as suggested in Section VI.

The classifier will also benefit from a training set with hands of many different shapes and grasped objects of different sizes. Although, this will increase the size of the database, the sub-linear computational complexity of the LSH approximate kNN search ensures that the computation time will grow at a very limited rate.

This paper discussed instantaneous recognition of grasps, recognized in isolation. Most probably, a higher recognition performance can be reached using a sequence of images over time. Moreover, there is a statistical correlation between types of objects, object shapes, human hand actions, and human grasps in a PbD scenario. We are therefore on our way to integrating the grasp classifier into a method for continuous simultaneous recognition of objects and human hand actions, using Conditional Random Fields (CRF) [28].

IX. Acknowledgments

This research has been supported by the EU through the project PACO-PLUS, FP6-2004-IST-4-27657, and by the Swedish Foundation for Strategic Research.

References

[1] S. Ekvall, "Robot task learning from human demonstration," Ph.D. dissertation, KTH, Stockholm, Sweden, 2007.

[2] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 799–822, 1994.

[3] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[4] A. Billard, "Imitation: A review," in *Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed., 2002, pp. 566–569.

[5] K. Ogawara, S. Iba, H. Kimura, and K. Ikeuchi, "Recognition of human task by attention point analysis," in *IEEE International Conference on Intelligent Robots and Systems*, 2000, pp. 2121–2126.

[6] M. C. Lopes and J. S. Victor, "Visual transformations in gesture imitation: What you see is what you do," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 2375–2381.

[7] S. Ekvall and D. Kragić, "Grasp recognition for programming by demonstration tasks," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 748–753.

[8] S. Calinon, A. Billard, and F. Guenter, "Discriminative and adaptative imitation in uni-manual and bi-manual tasks," in *Robotics and Autonomous Systems*, vol. 54, 2005.

[9] M. Cutkosky, "On grasp choice, grasp models and the design of hands for manufacturing tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 269–279, 1989.

[10] J. Rehg and T. Kanade, "Visual tracking of high dof articulated structures: An application to human hand tracking," in *European Conference on Computer Vision*, vol. 2, 1994, pp. 35–46.

[11] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, "A hand-pose estimation for vision-based human interfaces," in *IEEE Transactions on Industrial Electronics*, vol. 50(4), 2003, pp. 676–684.

[12] V. Athitsos and S. Sclaroff, "Estimating 3D hand pose from a cluttered image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 432–439.

[13] C. Schwarz and N. Lobo, "Segment-based hand pose estimation," in *Canadian Conf. on Computer and Robot Vision*, 2005, pp. 42–49.

[14] L. Y. Chang, N. S. Pollard, T. M. Mitchell, and E. P. Xing, "Feature selection for grasp recognition from optical markers," in *IEEE International Conference on Intelligent Robots and Systems*, 2007.

[15] H. Murase and S. Nayar, "Visual learning and recognition of 3-D objects from appearance," *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.

[16] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *International Conference on Very Large Databases*, 1999, pp. 518–529.

[17] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," in *International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, 1999, pp. 103–115.

[18] K. Ogawara, J. Takamatsu, K. Hashimoto, and K. Ikeuchi, "Grasp recognition using a 3D articulated model and infrared images," in *IEEE International Conference on Intellingent Robots and Systems*, vol. 2, 2003, pp. 1590–1595.

[19] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1372–1384, 2006.

[20] E. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, "Visual hand tracking using non-parametric belief propagation," in *IEEE Workshop on Generative Model Based Vision*, 2004.

[21] A. A. Argyros and M. I. A. Lourakis, "Real time tracking of multiple skin-colored objects with a possibly moving camera," in *European Conference on Computer Vision*, vol. 3, 2004, pp. 368–379.

[22] W. T. Freeman and M. Roth, "Orientational histograms for hand gesture recognition," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 1995.

[23] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter sensitive hashing," in *IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 750–757.

[24] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: a texture classification example," in *IEEE International Conference on Computer Vision*, vol. 1, 2003, pp. 456–463.

[25] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *Journal of Machine Learning Research*, no. 6, pp. 363–392, 2005.

[26] J. Tegin, S. Ekvall, D. Kragić, B. Iliev, and J. Wikander, "Demonstration based learning and control for automatic grasping," in *International Conference on Advanced Robotics*, 2007.

[27] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Neural Information Processing Systems*, 2006.

[28] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning*, 2001.

Simultaneous Visual Recognition of Manipulation Actions and Manipulated Objects

Hedvig Kjellström Javier Romero David Martínez Danica Kragić

Computational Vision and Active Perception Lab
School of Computer Science and Communication
KTH, SE-100 44 Stockholm, Sweden
hedvig,jrgn,davidmm,dani@kth.se

Abstract. The visual analysis of human manipulation actions is of interest for e.g. human-robot interaction applications where a robot learns how to perform a task by watching a human. In this paper, a method for *classifying manipulation actions in the context of the objects manipulated*, and *classifying objects in the context of the actions used to manipulate them* is presented. Hand and object features are extracted from the video sequence using a segmentation based approach. A shape based representation is used for both the hand and the object. Experiments show this representation suitable for representing generic shape classes. The action-object correlation over time is then modeled using conditional random fields. Experimental comparison show great improvement in classification rate when the action-object correlation is taken into account, compared to separate classification of manipulation actions and manipulated objects.

1 Introduction

Manipulation actions, i.e. hand actions for picking up objects, doing something with them and putting them down again, is an important class of hand activity not well studied in computer vision. The analysis of human manipulation is of interest for work-flow optimization, automated surveillance, and programming by demonstration (PbD) applications, in which a robot learns how to perform a task by watching a human do the same task.

An important cue to the class of a manipulation action is the object handled; for example, seeing a human bring a cup towards his/her face brings us to believe that he/she is drinking, without actually seeing the fluid. Similarly, a strong cue to the class of the object involved is the action; for example, a cup is to some extent defined as something you drink from. Therefore, it is beneficial to *simultaneously* recognize manipulation actions and manipulated objects.

A manipulation action is here defined as beginning with the picking-up of an object and ending with the putting-down of that object. Only one-hand actions are considered, although this is not a limitation to the method in a formal sense. In a video sequence of the action, the human head and hand are segmented and

tracked using skin color, and objects are segmented as being in the neighborhood of the hand and moving with it.

The action state space in each frame is the image position of the hand relative to the face, and the shape of the hand, represented with a gradient orientation histogram pyramid [1, 2]. Section 5 shows the inclusion of hand shape in the state space to greatly improve action recognition compared to only hand position.

Objects in this application are "graspable" i.e. fairly rigid, so shape is a good object descriptor. We use pyramids of gradient orientation histograms for representation of object shape as well as hand shape. Experiments in Section 5 show this representation to lead to a state-of-the-art classification performance on the NORB dataset [3] which contains objects of this type. Specific for our application is that the classification method has access to several views of the object over the course of the action, something that improves the recognition. Section 3 describes the feature extraction.

There are implicit, complex interdependencies in the object and action data. The sequence of object viewpoints, as well as occlusion from the hand depend on the action; i.e. what the hand is doing with the object. Similarly, the hand shape depends on the size, shape and surface structure of the object in the hand. These dependencies are difficult to model, which leads us to use a discriminative sequential classifier, conditional random fields (CRF) [4], that does not model the data generation process.

On a semantic level, there are also action-object dependencies of the type "drink"–"cup", "drink"–"glass", "write"–"pen", "draw"–"pen" and so on, which can be explicitly modeled within the CRF framework. The action-object dependence can be modeled on a per-frame basis using a factorial CRF (FCRF) [5]. However, it might be the case that the dependencies between particular frames are weaker than the dependence beween the action and the object as whole. To model sequence-level dependence, we introduce a CRF stucture which we call *connected hierarchic CRF:s (CHCRF)*. This is detailed in Section 4.

Experiments in Section 5 show three things. Firstly, CRF structures with many degrees of freedom, such as structures with hidden nodes or large data connectivity, perform worse than simple structures when the amount of training data is limited. Secondly, the correlated action-object recognition outperform separate classification, and CHCRF:s perform better than FCRF:s on the action-object classification task. Last, the information on actions implicit in the object data is redundant to the information on objects in the action data.

The primary contribution of this paper is the idea of recognizing manipulation actions and manipulated objects in context of each other, while secondary contributions are the definition of the CHCRF and the representation of object shape using pyramids of gradient orientation histograms.

2 Related Work

Actions. In the last few years, considerable research effort has been spent on the analysis of human motion from video [6]. For the purpose of detecting atomic

actions in video, Laptev and Pérez [7] use a boosted classifier of spatiotemporal volumes of optical flow. This approach is robust to significant changes in scale, appearance and viewpoint. Our method differs from [7] in that it is possible to model actions with a longer extension in time, possibly with a sub-structure. Furthermore, since our analysis involves the manipulated object, the hand needs to be located, putting different constraints on our feature extraction.

The analysis of hand motion is most often applied to gesture recognition for human-computer interfaces or sign language recognition [8]. These applications are often characterized by low or no occlusion of the hands from other objects, and a well defined and visually disparate set of hand poses; in the sign language case the gestures are designed to be easily separable to simplify fast communication. In contrast, the manipulation actions which we investigate suffer from large intra-class variability and sometimes occlusion of parts of the hand from the manipulated objects.

Feature extraction for hand action classification often means tracking the hand in 2D [9] or 3D [10, 11]. However, to be able to handle low video framerates we prefer to use a segmentation-based method for human pose recovery not relying on time-incremental estimation [12–14].

Objects. Object recognition is a vast area of research and can be regarded as one of the core problems in computer vision. We do not make an attempt to review the whole field, but focus on contextual object recognition and the representation of shape in object recognition.

The caption of an image says something about what objects can be expected in it. When labeling images according to object content, any captions should therefore be taken into account. Caption-guided object detection can be used to segment the image into object regions and associate them with object labels [15], or to automatically label or cluster a large set of unlabeled images with captions given a smaller set of labeled images with captions [16].

In [17–19], the scene itself, the "gist" of the image, is used to guide object recognition. The scene itself is a strong prior cue as to which objects can be expected and where they are most likely to be found. CRF:s have also been used [19, 20] to automatically learn sub-structure; the relations between different parts of the object or between different objects and the scene.

Earlier work on contextual object recognition [21, 22] has focused on functional object recognition; objects are then classified in terms of their function. This is similar in spirit to our contextual recognition; object classes are here defined by how the objects are used (in which action context they appear), and classes of manipulation actions are defined by the class (or classes) of objects that are involved in the action.

Modeling shape is difficult; an important tradeoff is the sparseness of the representation (from silhouettes [13], via edge maps [14, 23], to maps of gradient orientation) versus the robustness towards differences in lighting and fine object texture. We use pyramids of localized histograms of gradient orientation, a representation robust to small position and fine texture differences, while containing more texture information than e.g. edge maps.

Actions and objects. Moore et al. [24] provide a Bayesian framework for recognizing scenes, objects in it, and actions being performed on the objects. A system such as this could be the framework for our method: While their system keeps track the hands of a human, objects handled by the human, and which actions are performed on which objects, our method is concerned with classifying a single action and object.

Wu et al. [25] continue along this line, learning a dynamic Bayesian network model that represent *temporal sequences* of actions and objects involved in the actions. The features used for classification are RFID tags attached to the objects and the human hand. Again, our method could be incorporated in such a system, replacing the RFID-based classificiation with simultaneous classification of objects and actions from video.

Earlier on, Mann and Jepson [26] use a force-dynamic bottom-up approach to describe the interaction between hands and objects in scenes. In contrast, we use a statistical formulation, since the underlying process generating the video sequences in our case is far too complex to be modeled deterministically.

3 Features for Classification

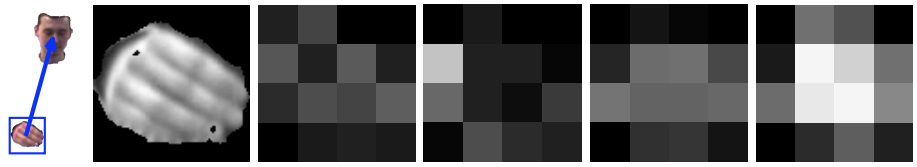
Extraction of image features could be done in a variety of ways [6] depending on the purpose of the feature extraction. Features representing the human motion and appearance as well as object motion and appearance are of interest. As opposed to many other action recognition applications, like video annotation [7], it is here necessary to obtain the location of the human hand to find the manipulated object.

Considering the low framerate, large motion blur and low resolution of the hands of our video sequences, and that articulated hand tracking is a difficult problem [10, 11], we use a segmentation based approach. The hand and face of the human is localized using skin color segmentation [27] and hand and face masks are extracted from the skin mask using connected components detection or with an $\alpha\beta$-filter when hand and face blobs merge.

Other cues than skin color, e.g. combinations of spatial or spatiotemporal filters, can of course also be exploited for the localization of hands and face.

The object involved in the manipulation action is in the human's right hand. To focus the attention of the object classification onto only that object, an object segmentation mask is also obtained, right of the hand in the image (based on the assumption that the object is in that area if grasped by the human). While the position of the area is automatically obtained from the hand position, the area shape is selected to fit the object in the first frame of the sequence, and is then held constant throughout the sequence.

Manipulation actions are here defined as beginning with a pick-up event and ending with a put-down event. With a fully automatic object segmentation, it is possible in each time-step to detect whether there is an object in the hand or not, so that the temporal segmentation can be done automatically.



(a) Pos (b) Image (c) $\pi/8$ (d) $3\pi/8$ (e) $5\pi/8$ (f) $7\pi/8$

Fig. 1. Features used for action classification. a) 2D position p_t of the centroid of the right hand segment relative to centroid of the face segment. b) Hand image $\mathbf{I}_t^a$. c–f) Gradient orientation histograms from $\mathbf{I}_t^a$, with $B = 4$ bins, on level $l = 1$ in the pyramid of $L = 3$ levels. c) Bin 1, orientation $\pi/8$. d) Bin 2, orientation $3\pi/8$. e) Bin 3, orientation $5\pi/8$. f) Bin 4, orientation $7\pi/8$.

Action features. We seek a representation of hand motion that captures both the global hand position and the articulated hand pose over time. The global position of the hand at time t is represented with the 2D position p_t^a of the centroid of the hand mask relative to the centroid of the face mask (Figure 1a).

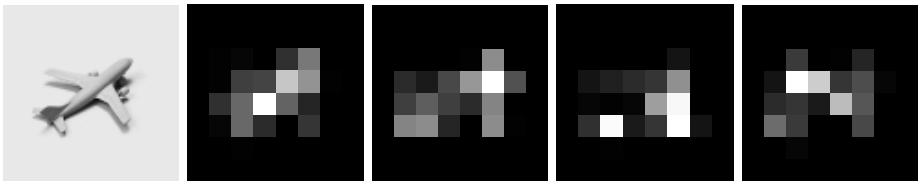
The local articulated hand pose is represented using gradient orientation histograms, frequently used for representation of human shape [1, 2]. Gradient orientation $\boldsymbol{\Phi}_t \in [0, \pi)$ is computed from the segmented hand image $\mathbf{I}_t^a$ (Figure 1b) as $\boldsymbol{\Phi}_t = \arctan(\frac{\partial \mathbf{I}_t^a}{\partial y} / \frac{\partial \mathbf{I}_t^a}{\partial x})$ where x denotes downward (vertical) direction and y rightward (horizontal) direction in the image.

From $\boldsymbol{\Phi}_t$, a pyramid with L levels of histograms with different spatial resolutions are created; on each level l, the gradient orientation image is divided into $2^{L-l} \times 2^{L-l}$ equal partitions. A histogram with B bins is computed from each partition. Figures 1c–f show histograms at the lowest level of the pyramid.

The hand pose at time t is represented by the vector x_t^a which is the concatenation of the position p_t^a and all histograms at all levels in the pyramid. The length of x_t^a is thus $2 + B \sum_{l=1}^{L} 2^{2(L-l)}$. The performance of the classifier is quite insensitive to choices of $B \in [3, 8]$ and $L \in [2, 4]$; in our experiments in Section 5 we use $B = 4$ and $L = 3$. Before concatenation, p_t^a is normalized so that the standard deviations of the two dimensions of x_t^a originating from p_t^a have the same standard deviation in the training set (Section 5) as the remaining dimensions. The sequence of poses over the sequence is $\mathbf{x}^a = \{x_t^a\}, t = 1, \ldots, T$.

Object features. The objects considered in this application are all "graspable", i.e. more or less rigid. For example, a cup is graspable, but water is not. Shape can therefore be expected to be a good object class descriptor, while local descriptors like SIFT features [28] are unsuitable for our purposes.

Contrary to in many other object recognition applications, e.g. labeling of images according to object content, there is no search for object location involved. For manipulated objects, the position of the hand grasping the object gives an indication of the expected object location, and the recognition problem becomes one of classifying the given region. However, there might be deviations in position and orientation of the object within this region, as well as devia-



(a) Image (b) $\pi/8$ (c) $3\pi/8$ (d) $5\pi/8$ (e) $7\pi/8$

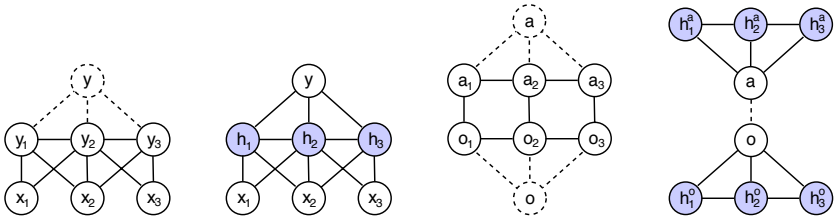
Fig. 2. Features used for object classification. a) Object image $\mathbf{I}_t^o$. b–e) Gradient orientation histograms from $\mathbf{I}_t^o$, with $B = 4$ bins, on level $l = 1$ in the pyramid of $L = 4$ levels. b) Bin 1, orientation $\pi/8$. c) Bin 2, orientation $3\pi/8$. d) Bin 3, orientation $5\pi/8$. e) Bin 4, orientation $7\pi/8$.

tions in size, shape and color of different instances within each object class. The descriptor must therefore be insensitive to these intra-class variations, but still capture inter-class variations.

We select the same gradient orientation histogram representation as for the hand shape, a description that captures the shape of the object, with a certain insensitivity to absolute greylevels and small displacements of object parts (Figure 2). Note that this representation is not invariant to e.g. in-plane rotations; this is deliberate, since global orientation is indicative of object class in our application. The object at time t is represented by x_t^o, the concatenation of all histograms at all levels in the pyramid. In Section 5 this representation is evaluated on the problem of recognizing generic object categories, and it is found to be robust to intra-class variability in shape, orientation, position, rotation and lighting conditions, while maintaining a good inter-class discriminability.

Another factor specific to this object recognition application is that the data consist of not only one, but a sequence of object views. In most cases, parts of the object are also occluded by the human hand grasping it. The change in orientation of the object with respect to the camera during the sequence and the occlusion from the hand are descriptive of the object, since they reflect the way this object class is used by the human; they can be termed "typical view sequences" and "typical occlusions". Thus, the classifier should take the whole sequence of object views into account. Each measurement is therefore described by a sequence of descriptors $\mathbf{x}^o = \{x_t^o\}, t = 1, \ldots, T$.

Correlation between action and object features. As discussed in the introduction, the shape of the hand encoded in $\mathbf{x}^a$ gives cues about the object as well, since humans grasp different types of objects differently, due to object function, shape, weight and surface properties. Similarly, the view change in $\mathbf{x}^o$ over the course of the sequence is correlated with the type of action performed with the object. This representation of the correlation between manipulation actions and manipulated objects is *implicit* and difficult to model accurately, but should be taken into account when modeling the simultaneous action-object recognition.



(a) CRF (b) HCRF (c) FCRF (d) CHCRF

Fig. 3. Different CRF structures used for action-object recognition with pre-segmented data. Dotted edges and nodes indicate that the weights θ associated with these are constrained during the training of the CRF. a) Linear-chain CRF [4]. b) Hidden CRF [20]. c) Factorial CRF [5], data layer not shown. d) Connected hierarchic CRF:s, data layer not shown.

4 Discriminative Classification using CRF:s

Since we can expect complex dependencies within our action data $\mathbf{x}^a$ and object data $\mathbf{x}^o$ over time, a discriminative classifier which does not model the data generation process is preferable over a generative sequential classifier like a hidden Markov model (HMM) [29]. We thus employ conditional random fields (CRF:s) [4] which are undirected graphical models that represent a set of state variables $\mathbf{y}$, distributed according to a graph $\mathcal{G}$, and conditioned on a set of measurements $\mathbf{x}$. Let $C = \{\{\mathbf{y}_c, \mathbf{x}_c\}\}$ be the set of cliques in $\mathcal{G}$. Then,

$$P(\mathbf{y}|\mathbf{x};\theta) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Phi(\mathbf{y}_c, \mathbf{x}_c; \theta_c) \tag{1}$$

where Φ is a potential function parameterized by θ as

$$\Phi(\mathbf{y}_c, \mathbf{x}_c; \theta_c) = e^{\sum_k \theta_{c,k} f_k(\mathbf{y}_c, \mathbf{x}_c)} \tag{2}$$

and $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in C} \Phi(\mathbf{y}_c, \mathbf{x}_c; \theta_c)$ is a normalizing factor. The feature functions $\{f_k\}$ are given, and training the CRF means setting the weights θ using belief propagation [4].

For linear-chain data (for example a sequence of object or action features and labels), $\mathbf{y} = \{y_t\}$ and $\mathbf{x} = \{x_t\}$, $t = 1, \ldots, T$ as shown in Figure 3a. This means that the cliques are the edges of the model, which gives

$$P(\mathbf{y}|\mathbf{x};\theta) = \frac{1}{Z(\mathbf{x})} \prod_{t=2}^{T} \Phi(y_{t-1}, y_t, \mathbf{x}; \theta_t) \tag{3}$$

with a potential function

$$\Phi(y_{t-1}, y_t, \mathbf{x}; \theta_t) = e^{\sum_k \theta_{t,k} f_k(y_{t-1}, y_t, \mathbf{x})}. \tag{4}$$

Each state y_t can depend on the whole observation sequence $\mathbf{x}$ – or any subpart of it, e.g. the sequence $\{x_{t-\mathcal{C}}, \ldots, x_{t+\mathcal{C}}\}$, $\mathcal{C}$ being the *connectivity* of the model.

A CRF returns an individual label for each time-step, which means that the time-sequential data $\mathbf{x}$ to be classified do not have to be segmented prior to classification. However, if a segmentation is readily available, as in our case, the robustness of the classification is increased by assuming that for all labels within the segment, $y_t = y \,\forall\, t \in [1, T]$. With pre-segmentation, the model thus becomes $P(y|\mathbf{x};\theta) \equiv P(\mathbf{y} = [y, \ldots, y]|\mathbf{x};\theta)$. This is illustrated by the dotted layer in Figure 3a. We will formalize this below.

The introduction of a hidden layer, where each hidden label represents the classification of a sub-part of the sequence (Figure 3b), has been shown [20] to improve the recognition rate in situations where there is such a sub-structure present in the data. This is indeed the case in our object and action recognition problems. With an observable label y and a set of hidden labels $\mathbf{h}$ that form a time-chain, the probabilistic model of a hidden CRF (HCRF) becomes

$$P(y|\mathbf{x};\theta) = \frac{1}{Z(\mathbf{x})} \sum_{\mathbf{h}} \prod_{t=1}^{T} \Phi(y, h_t, \mathbf{x}; \theta_t^h) \prod_{t=2}^{T} \Phi(y, h_{t-1}, h_t, \mathbf{x}; \theta_t^{hh}) \tag{5}$$

with potential functions

$$\Phi(y, h_t, \mathbf{x}; \theta_t^h) = e^{\sum_k \theta_{t,k}^h f_k(y, h_t, \mathbf{x})},\ \Phi(y, h_{t-1}, h_t, \mathbf{x}; \theta_t^{hh}) = e^{\sum_k \theta_{t,k}^{hh} f_k(y, h_{t-1}, h_t, \mathbf{x})}. \tag{6}$$

Both inference and parameter estimation can be done using exact methods, provided that there are no loops in the hidden layer [20]. However, the introduction of hidden parameters leads to a non-convex optimization problem, which means that the parameter estimation procedure requires more data to converge, and might also reach local optima. Note also that an HCRF requires pre-segmented data; for continuous classification, other structures with hidden layers have been presented, like the latent-dynamic CRF (LDCRF) [30]. CRF and HCRF performance for pre-segmented data is compared in the experiments in Section 5.

The CRF in Figure 3a is in fact a special case of the HCRF in Figure 3b, where the weights θ are restricted so that *A*) $\mathbf{h}$ are not hidden, $h_t = y \,\forall\, t \in [1, T]$; and *B*) equal weight is given to each timestep, $\theta_{t_1}^h = \theta_{t_2}^h \,\forall\, t_1, t_2 \in [1, T]$.

"Early fusion": factorial CRF. In Section 3 we argue that there are correlations between action observations $\mathbf{x}^a$ and object observations $\mathbf{x}^o$ implicit in the data. We make use of this correlation on the data level by not imposing a simplified model on the data generation process and instead using a discriminative classifier, CRF. However, there is also an *explicit*, semantic correlation between actions and objects on the label level, as discussed in the introduction. This correlation can be modeled in two ways, which we denote "early" and "late fusion". Early fusion corresponds to modeling the correlation on a per-frame basis, i.e. the correlations between the labels a_t and o_t for each frame of the action, using a factorial CRF (FCRF) [5]. Figure 3c shows an FCRF with two states,

action class a_t and object class o_t, in each time-step t. The conditional dependence on data is omitted in the figure for visibility. The cliques in this model are the within-chain edges $\{a_{t-1}, a_t\}$ and $\{o_{t-1}, o_t\}$, and the between-chain edges $\{a_t, o_t\}$. The probability of $\mathbf{a}$ and $\mathbf{o}$ is thus defined as

$$P(\mathbf{a}, \mathbf{o}|\mathbf{x}; \theta) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \Phi(a_t, o_t, \mathbf{x}; \theta_t) \prod_{t=2}^{T} \Phi(a_{t-1}, a_t, \mathbf{x}; \theta_{a,t}) \Phi(o_{t-1}, o_t, \mathbf{x}; \theta_{o,t}) \ . \tag{7}$$

The weights θ are obtained during training using loopy belief propagation [5]. As for the linear-chain CRF, pre-segmentation means that $a_t = a, o_t = o \forall\, t \in [1, T]$ and that the distribution over the two dotted layers in the FCRF can be expressed as $P(a, o|\mathbf{x}; \theta) \equiv P(\mathbf{a} = [a, \ldots, a], \mathbf{o} = [o, \ldots, o]|\mathbf{x}; \theta)$.

"Late fusion": connected hierarchic CRF:s. Late fusion corresponds to modeling the correlation on a sequence level. The assumption is here that it is the action label a that is correlated with the object label o, not the labels a_t and o_t of a particular frame. The structure of a CRF for "late fusion", called *connected hierarchic CRF:s (CHCRF)* is shown in Figure 3d. In the most general case the two linear-chain layers are hidden, and the probability of an action a and an object o conditioned on the data $\mathbf{x}$ is

$$\begin{aligned} P(a, o|\mathbf{x}; \theta) = \frac{1}{Z(\mathbf{x})} \Phi(a, o, \mathbf{x}; \theta^{ao}) & \sum_{\mathbf{h}^a} \prod_{t=1}^{T} \Phi(a, h_t^a, \mathbf{x}; \theta_t^a) \prod_{t=2}^{T} \Phi(a, h_{t-1}^a, h_t^a, \mathbf{x}; \theta_t^{aa}) \\ & \sum_{\mathbf{h}^o} \prod_{t=1}^{T} \Phi(o, h_t^o, \mathbf{x}; \theta_t^o) \prod_{t=2}^{T} \Phi(a, h_{t-1}^o, h_t^o, \mathbf{x}; \theta_t^{oo}) \\ = \frac{\Phi(a, o, \mathbf{x}; \theta^{ao})}{\sum_{a,o} \Phi(a, o, \mathbf{x}; \theta^{ao})} & P(a|\mathbf{x}; \theta^a) P(o|\mathbf{x}; \theta^o) \end{aligned} \tag{8}$$

in analog with Eq (5). To make the training more efficient, the data dependency in the first term is omitted, becoming $\mathcal{K}(a, o) = \frac{\Phi(a,o;\theta^{ao})}{\sum_{a,o} \Phi(a,o;\theta^{ao})}$, the *co-occurrence rate* of action label a and object label o in the training data. The individual probabilities $P(a|\mathbf{x}; \theta^a)$ and $P(o|\mathbf{x}; \theta^o)$ are estimated as in Eq (3), Eq (5), or using any classification method that returns probability estimates. The parameters of the action and object classifiers are learned separately.

5 Experiments

Evaluation of the object features. The object feature representation was first evaluated on its own, without the CRF framework. For this we used the NORB dataset [3], which contains 5 different classes of rigid objects; "animals", "humans", "airplanes", "trucks", and "cars" with 10 instances of each, 5 for test and 5 for training.The database contains stereo views of each object from 18 different

azimuths and 9 elevations in 6 different lighting conditions. Only the normalized-uniform part of the dataset, designed to test classification performance, was used. (The other part of the dataset is designed to test object detection, a task we do not claim to address with this method.)

To evaluate the suitability of the feature representation for modeling shape *categories*, a support vector machine (SVM) [31] was trained with feature vectors x^o extracted from the NORB training images as described in Section 3. Table 1 left shows the results compared to others. Our object view representation together with a standard classifier reached the same classification accuracy as a state-of-the-art method for object categorization [3], which indicates that the gradient orientation histograms capture the specifics of a shape class, while allowing a significant variability among instances of that class. In comparison, training on the raw image downsampled to a size of 32×32 led to twice the classification error (a surprisingly good result, as noted in [3], given that the task is object categorization, not instance recognition). Furthermore, we note that the incorporation of stereo does not add much to the accuracy.

Certain robustness towards differences in color and lighting, as well as small position errors of the object segmentation mask, is also desireable. In [3], this was tested by adding "jitter", i.e. small transformations to both the training and test set. However, this arguably tested how the methods performed with a larger test set, rather than how they could handle noise that was not seen before (not present in the training data). Therefore, we did a variant of this experiment where we added jitter to *only the test set* (Table 1 right). First, the overall brightness of each test image was varied. Our feature representation was very robust to this noise, which is expected since it relies solely on the gradient orientations and not on their value. In comparison, the raw image classification error grew much quicker. Then, the test images were shifted vertically and horizontally in a random manner. The feature representation was more sensitive to this noise, but less so than the raw image representation.

Simultaneous action and object recognition. For the purpose of testing the simultaneous action-object recognition, the OAC (Object-Action-Complex) dataset was collected. The dataset consists of 50 instances each of three different action-object combinations; "look through binoculars", "drink from cup", and "pour from pitcher". The actions were performed by 7 men and 3 women, 5 times each. The classes were selected so that the object and action data are complementary:

Table 1. Results on the normalized-uniform NORB dataset, percent error. Left: Classification error percentage compared to methods presented in [3]. Right: Generalization test; robustness to different amount of jitter in test data (training data unaltered).

	Mono	**Stereo**
Hist + SVM	6.4	6.2
Raw + SVM	12.6 **[3]**	—
Conv Net 80	—	6.6 **[3]**

		Brightness			**Shift**		
	±0	**±10**	**±20**	**±30**	**±3**	**±6**	**±9**
Hist + SVM	6.4	6.4	7.1	8	10.3	18.1	29.2
Raw + SVM	12.6 **[3]**	15.8	18	21	20.8	35.1	48.6

(a) Instance of "look through binoculars"

(b) Instance of "drink from cup"

(c) Instance of "pour from pitcher"

Fig. 4. The three classes of the OAC dataset (for one person, instance each). Training and testing was performed in a jackknife manner, where the 15 sequences of one person at a time was used as a test data, the 135 sequences of the other 9 persons as training set.

two of the actions, "look through" and "drink from" are similar, while "cup" and "pitcher" are similar.

Only one instance of each object was used, so the full object representation generated a perfect classification performance with all parameter settings. To simulate the performance in the more general object category recognition case, all spatial information was removed, by using only $L = 1$ level in the gradient orientation histogram pyramid, and by normalizing the object segmentation window with respect to aspect ratio (i.e. scale all object segments to squares). The experiments below are not indicative of the object classification, but rather of the action classification and the benefits of combining object and action classification for manipulation action applications.

Table 2. Experiments with separate action and object (H)CRF classification with different connectivity $\mathcal{C}$, percent error on the format "median (max)" of 9 runs.

	CRF conn 0	**CRF conn 1**	**CRF conn 2**	**HCRF conn 0**	**Baseline**
Actions	5.3 (9.3)	6 (12.7)	10.7 (18.7)	14.7 (21.3)	17.3 (20) **(2D)**
Objects	8 (8.7)	11.3 (13.3)	20.7 (28)	24.7 (28.7)	8.7 **(1:st fr)**

Table 2 shows separate action and object classification with different parameter settings. The baseline for actions is a representation without spatial information – only 2D pos $\mathbf{p}^a$ over time, and for objects a representation without temporal information – only the first frame x_1^o. Two things can be noted. Firstly, while the object classification does well without the temporal information, the action is to a large degree determined by spatial information, i.e. the shape of the hand. Secondly, when comparing to similar experiments in [20] where HCRF:s outperformed CRF:s, we draw the conclusion that it depends on the amount of training data. With relatively little data, a model with fewer parameters will perform better. In the applications we are considering, the less training data needed, the better, since data collection takes time and the system should be adaptable to different environments.

When comparing late and early fusion (Table 3) we see that while late fusion (CHCRF) greatly improves the classification, there is only a marginal improvement in the classification with early fusion (FCRF). There are two possible interpretations which might both be true: *A*) the per-frame correlation of actions and objects is simply a bad model of reality; *B*) the larger set of parameters defining the FCRF leads to a worse performance with our relatively small training set. As a side note, the CHCRF object and action classification rates are identical since the co-occurrence matrix $\mathcal{K}$ (Eq (8)) is diagonal in this example. A non-unique action-object mapping (e.g. "drink"–"glass" and "drink"–"cup") would lead to differences in action and object classification rate.

How important is the implicit, data-level correlation compared to the explicit, semantic correlation modeled in the FCRF and CHCRF? To test this, late fusion was performed with full data, $\mathbf{x} = [\mathbf{x}^a, \mathbf{x}^o]$, correlation in either cue removed, $\mathbf{x} = [\mathbf{x}^a, x_1^o]$ or $[\mathbf{p}^a, \mathbf{x}^o]$, and all correlation removed, $\mathbf{x} = [\mathbf{p}^a, x_1^o]$. From Table 4, it seems that the information on objects in the action data $\mathbf{x}^a$ and the information on actions in the object data $\mathbf{x}^o$ is largely redundant; the classification performance is not affected to any greater extent by using *either* $\mathbf{p}^a$ or x_1^o, but if *all* correlation data are removed, the classification is seriously affected.

6 Conclusions

A method for simultaneous sequential recognition of manipulation actions and manipulated objects was presented, employing CRF:s trained with object and hand shape over the course of the action. Two different CRF structures for fusion of action and object classification were compared; FCRF:s, which model the correlation on a per-frame level, and CHCRF:s, a structure introduced in this

Table 3. Experiments comparing late (CHCRF) and early (FCRF) fusion, percent error on the format "median (max)" of 9 runs. Connectivity 0 everywhere.

	Connected Hiearchic CRF:s	**Factorial CRF**	**Baseline (sep CRF:s)**
Actions	3.3 (4.7)	5.3 (8)	5.3 (9.3)
Objects	3.3 (4.7)	6 (12)	8 (8.7)

paper, which correlate the action and object classifcation as whole. CHCRF:s outperformed FCRF:s on the task of correlated classification of action and object sequences, probably because the action and object data are not correlated on a per-frame basis.

CRF structures with many degrees of freedom, such as HCRF:s, or CRF:s with high data connectivity, were found to perform worse than simpler structures with relatively small amounts of training data, although they have higher descriptive power. Thus, in applications where the training data are limited, the complexity of the model should be selected so that the training procedure will converge with the amount of training data at hand. Moreover, the implicit information on actions in the object data and on objects in the action data was found to be redundant. Thus, removing the correlated data in one cue or the other can be done without affecting the overall classification, while removing the correlated data in both cues will have serious effects on the classification rate.

The representation of object shape using a pyramid of gradient orientation histograms was shown to give state-of-the-art classification results on the NORB dataset, indicating that the representation is robust to intra-class differences in quite general shape categories, while capturing inter-class differences.

In the future, we plan to incorporate the method for correlated action and object classification into a PbD framework such as [24, 25].

Acknowledgments. This research has been supported by the EU through the project PACO-PLUS, FP6-2004-IST-4-27657, and by the Swedish Foundation for Strategic Research.

References

1. Freeman, W.T., Roth, M.: Orientational histograms for hand gesture recognition. In: IEEE Int. Conf. Automatic Face and Gesture Recognition. (1995)
2. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter sensitive hashing. In: ICCV. Volume 2. (2003) 750–757
3. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. In: CVPR. Volume 2. (2004) 97–104
4. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML. (2001)
5. Sutton, C., Rohanimanesh, K., McCallum, A.: Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In: ICML. (2004)

Table 4. How important is the implicit, data-level correlation to the overall action-object classification? Classification where none, either, or both cues are stripped baseline versions, percent error on the format "median (max)" of 9 runs. Late (CHCRF) fusion, connectivity 0 everywhere.

	Actions	**Actions Baseline (only 2D pos)**
Objects	3.3 (4.7)	2.7 (8)
Objects Baseline (SVM 1:st fr)	4.7 (8.7)	14.7 (18)

6. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in computer vision-based human motion capture and analysis. CVIU **104** (2006) 90–126
7. Laptev, I., Pérez, P.: Retrieving actions in movies. In: ICCV. (2007)
8. Pavlovic, V.I., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. PAMI **19** (1997) 677–695
9. Yang, M., Ahuja, N., Tabb, M.: Extraction of 2d motion trajectories and its application to hand gesture recognition. PAMI **24** (2002) 1061–1074
10. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Model-based hand tracking using a hierarchical bayesian filter. PAMI **28** (2006) 1372–1384
11. Sudderth, E., Mandel, M.I., Freeman, W.T., Willsky, A.S.: Visual hand tracking using non-parametric belief propagation. In: IEEE Workshop on Generative Model Based Vision. (2004)
12. Sminchisescu, C., Kanujia, A., Metaxas, D.: Conditional models for contextual human motion recognition. CVIU **104** (2006) 210–220
13. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI **24** (2002) 509–522
14. Sullivan, J., Carlsson, S.: Recognizing and tracking human action. In: ECCV. Volume 1. (2002) 629–644
15. Carbonetto, P., de Freitas, N., Barnard, K.: A statistical model for general contextual object recognition. In: ECCV. Volume 1. (2004) 350–362
16. Quattoni, A., Collins, M., Darrell, T.: Learning visual representations using images with captions. In: CVPR. (2007)
17. Torralba, A.: Contextual priming for object detection. IJCV **53** (2003) 169–191
18. Murphy, K., Torralba, A., Freeman, W.T.: Using the forest to see the trees: A graphical model relating features, objects, and scenes. In: NIPS. (2003)
19. Torralba, A., Murphy, K., Freeman, W.T.: Contextual models for object detection using boosted random fields. In: NIPS. (2004)
20. Quattoni, A., Wang, S., Morency, L., Collins, M., Darrell, T.: Hidden conditional random fields. PAMI **29** (2007) 1848–1852
21. Rivlin, E., Dickinson, S.J., Rosenfeld, A.: Recognition by functional parts. CVIU **62** (1995) 164–176
22. Stark, L., Bowyer, K.: Generic Object Recognition using Form and Function. World Sci. Ser. Machine Perception and Artificial Intelligence; Vol. 10 (1996)
23. Jurie, F., Schmid, C.: Scale-invariant shape features for recognition of object categories. In: CVPR. Volume 2. (2004) 90–96
24. Moore, D.J., Essa, I.A., Hayes, M.H.: Exploiting human actions and object context for recognition tasks. In: ICCV. (1999)
25. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.M.: A scalable approach to activity recognition based on object use. In: ICCV. (2007)
26. Mann, R., Jepson, A.: Towards the computational perception of action. In: CVPR. (1998)
27. Argyros, A.A., Lourakis, M.I.A.: Real time tracking of multiple skin-colored objects with a possibly moving camera. In: ECCV. Volume 3. (2004) 368–379
28. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
29. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77** (1989) 257–286
30. Morency, L.P., Quattoni, A., Darrell, T.: Latent-dynamic discriminative models for continuous gesture recognition. In: CVPR. (2007)
31. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Visual Object-Action Recognition for Learning from Demonstration

Hedvig Kjellström Danica Kragic
Computational Vision and Active Perception Lab
Centre for Autonomous Systems
CSC, KTH, SE-100 44 Stockholm, Sweden
hedvig,dani@kth.se

***Abstract*— Robot learning of household tasks from human demonstration necessitates the recognition of human manipulation actions, as well as reasoning about the manipulated objects. To this end, this paper describes a method for 1) automatically segmenting human motion from a video of manipulation action, 2) spatially segmenting the manipulated objects based on depth and motion relative to the manipulating hand, 3) temporally segmenting the video according to "pick-up" and "put-down" events, and 4) modeling the action as a sequence of primitive actions, each with one object. The primitives are recognized using a previously developed method for simultaneous recognition of manipulation actions and manipulated objects.**

I. Introduction

We are interested in robot learning from demonstration [1] in household settings. The aim is for a household robot to learn how to perform a task, e.g. making dinner, by watching a human performing the task. The specifics of this environment is that the tasks to be learned involve manipulation of objects in the environment.

A crucial issue in learning manipulation tasks from demonstration is that the number of observations of the specific task is very small. This is usually addressed by modeling the task as a sequence of primitive actions, which can be learned beforehand. The task is then described in terms of these primitives [2]. In this paper we address the issue of learning primitive actions, and segmenting and recognizing them from an (unsegmented) video of a new combination of primitive actions.

Going back to Gibson [3], the concept of *affordances* is of importance for learning of manipulation tasks. According to the affordance theory, objects are defined by how they can be used, for example, a cup can be defined as "fillable" and a chair or stool as "sittable". Building upon this concept, objects and actions performed upon them can be viewed as an unseparable entity, an *object-action complex* (OAC) [4]. In earlier work [5] we, to this end, present a method for simultaneously recognizing manipulation actions and manipulated objects. Objects are thus recognized in context of how they are manipulated, and manipulation actions are recognized in context of what object is manipulated.

Building on this work, we here show how such an OAC classifier can be used by a robot to 1) learn a composite task by modeling it as a sequence of previously seen OAC primitives, 2) learn about objects in the scene by observing how they are manipulated.

Specifically, we have extended the method in [5] with a method for segmenting manipulated objects from the background. Using the spatial object segmentation it is possible to segment primitive actions temporally, a higly non-trivial issue in action recognition. An action is here defined as beginning with a pick-up event and ending with a put-down event. Since the object segmentation method can determine if there is an object in the hand or not, these events can be detected automatically and used as temporal segmentation points. This is described in Section III.

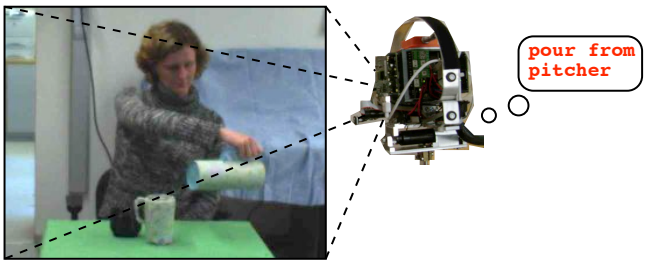


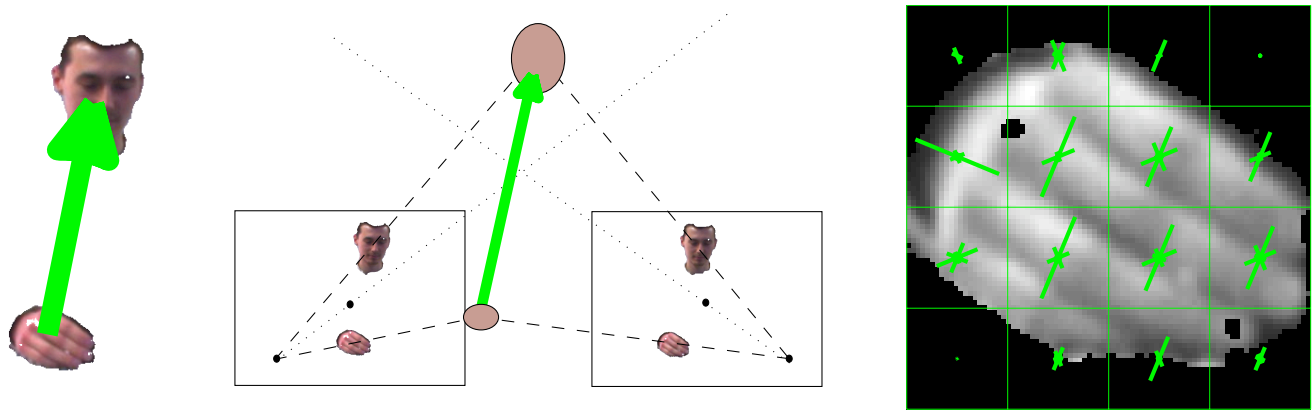
Fig. 1. Visual robot learning from demonstration. The robot models a human task as a sequence of primitive manipulation actions, classified simultaneouslty with the manipulated objects.

For completeness, we describe the OAC classification method from [5] in Section IV. Experiments in Section V investigate the benefits of extending the action classifier with 3D cues, and show how the classification method can be used to segment a sequence and recognize OACs in it.

II. Related Work

Within the robotics community there is a large interest in the recognition of human grasp activity for learning from demonstration applications. The 3D hand pose is typically recovered from 3D motion capture systems [6] or infrared images [7]. The hand pose is then classified from this representation. Instead, we use an implicit representation of hand shape, together with the global position of the hand in a human-centered coordinate system. For robot generation of the recognized action, the hand shape can be reconstructed from the implicit shape representation [8].

In the last few years, considerable research effort has been spent on the analysis of human motion from video [9]. The analysis of hand motion is most often applied to gesture recognition for human-computer interfaces or sign language recognition [10]. These applications are often characterized by low or no occlusion of the hands from other objects, and a well defined and visually disparate set of hand poses; in the sign language case the gestures are designed to be easily



(a) 2D hand-face position (b) 3D hand-face position (c) HOG over hand

Fig. 2. Features used for action classification. a) 2D position p_t of the centroid of the right hand segment relative to centroid of the face segment. a) 3D position q_t of the centroid of the right hand segment relative to centroid of the face segment, triangulated from stereo pair. b) Hand image $\mathbf{I}_t^a$, HOG superimposed.

separable to simplify fast communication. In contrast, the manipulation actions which we investigate suffer from large intra-class variability and sometimes occlusion of parts of the hand from the manipulated objects.

Feature extraction for hand action classification often means tracking the hand in 2D [11] or 3D [12], [13]. However, to be able to handle low video frame-rates we prefer to use a segmentation-based method for human pose recovery not relying on time-incremental estimation.

Building upon Gibson's affordance theory [3], in functional object recognition [14], [15] objects are modeled in terms of their function. This is similar in spirit to our contextual OAC recognition; object classes are here defined by how the objects are used (in which action context they appear), and classes of manipulation actions are defined by the class (or classes) of objects that are involved in the action. Similarly, Mann and Jepson [16] use a force-dynamic bottom-up approach to describe the interaction between hands and objects in scenes. In contrast, we use a statistical formulation, since the underlying process generating the sequences in our case is too complex to be modeled deterministically.

Moore et al. [17] provide a Bayesian framework for recognizing scenes, objects in it, and actions being performed on the objects. A system such as this could be the framework for our method, empoying the temporally segmented and classified OACs as primitives in the graphical structure. Wu et al. [18] continue along this line, learning a dynamic Bayesian network model that represent *temporal sequences* of actions and objects involved in the actions. The features used for classification are RFID tags attached to the objects and the human hand. Again, our method could be incorporated in such a system, replacing the RFID-based classificiation with classification of OACs from video.

III. Features for Classification

For our purposes, extraction of image features could be done in a variety of ways [9] depending on the purpose of the feature extraction. Features representing the human motion and appearance as well as object motion and appearance are of interest. As opposed to many other action recognition applications, it is here necessary to obtain the location of the human hand to find the manipulated object.

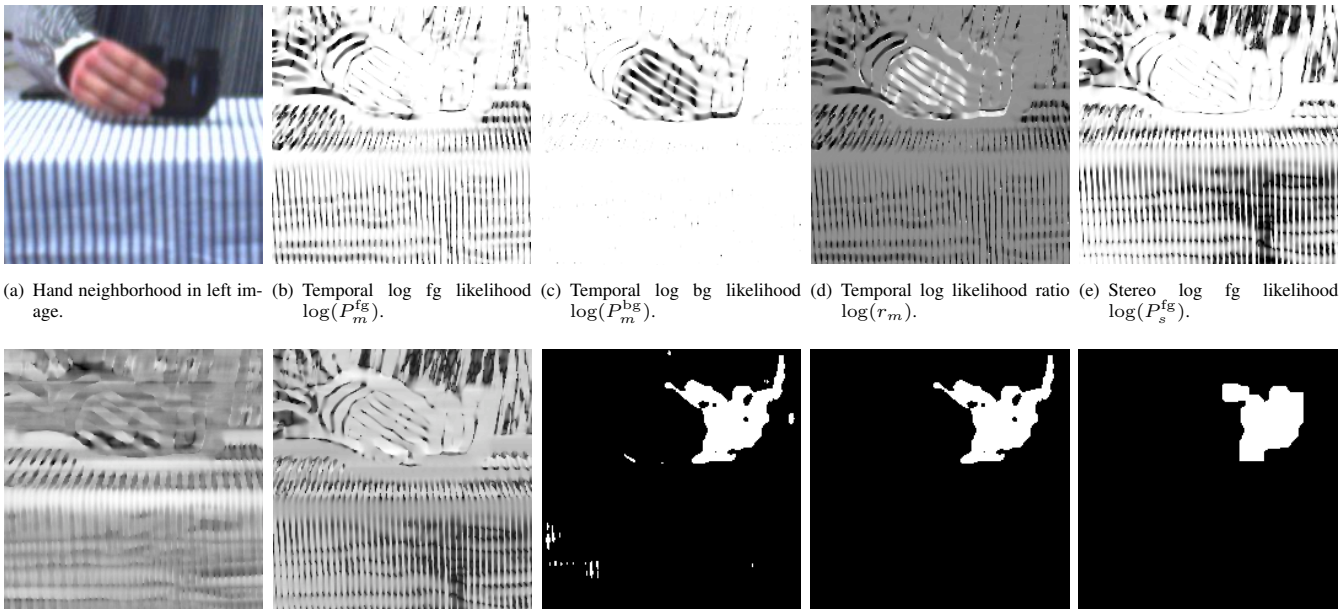
Furthermore, it should be possible to recreate the recognized action with a robot, which means that the hand 3D position, orientation and articulated pose should be retrievable from the action representation. This is further discussed in Section III-A below.

The observations consist of small baseline stereo video from two AVT Marlin F-080C cameras. Considering the low framerate, large motion blur and low resolution of the video sequences, we use a segmentation based approach. This is the same feature extraction approach as in [5], with the addition of hand position estimation in 3D and automatic object segmentation.

A. Action Features

The hand and face of the human is localized using skin color segmentation [19] and hand and face masks are extracted from the skin mask using connected components detection or with an $\alpha\beta$-filter when hand and face blobs merge. This is done separately for the left and right images $\mathcal{I}_t^L$ and $\mathcal{I}_t^R$ at each time instance t.

We seek a representation of human motion that enables reproduction of the action in another embodiment, for example a robot. We do not here address the problem of mapping motion from one embodiment to another. However, it can be noted that in order to be able to map a manipulation action on a task-level [1] the outcome of the action must be represented in the feature space. Eventually, modeling the outcome of manipulation actions necessitates modeling of object state over time (such as whole/chopped onion). We here assume a simplified representation of the world where the outcome of an manipulation action is captured by the motion of the manipulated object (and the hand grasping the object) over the course of the action. This means that the action becomes reproducable if the *global position* and



(a) Hand neighborhood in left image. (b) Temporal log fg likelihood $\log(P_m^{\mathrm{fg}})$. (c) Temporal log bg likelihood $\log(P_m^{\mathrm{bg}})$. (d) Temporal log likelihood ratio $\log(r_m)$. (e) Stereo log fg likelihood $\log(P_s^{\mathrm{fg}})$.

(f) Stereo log bg likelihood $\log(P_s^{\mathrm{bg}})$. (g) Stereo log likelihood ratio $\log(r_s)$. (h) Thresholding on $r_m r_s$. (i) Selecting connected region from (h). (j) Morphologial operations on (i).

Fig. 3. Segmentation of manipulated object. The hand has been segmented beforehand [8]. a) Hand region in left image of a stereo pair. The prior information about a manipulated object is that it is to be found in this region. b) Phase difference between hand region at times t and $t-1$ given the background hypothesis: background static. c) Phase difference between hand region at times t and $t-1$ given the foreground hypothesis: object moved with hand. d) The logarithm of the fg-bg temporal likelihood ratio, $\log(r_m)$. e) Phase difference between hand region in left and right stereo images given the background hypothesis: background on any depth. f) Phase difference between hand region in left and right stereo images given the foreground hypothesis: object on same depth as hand. g) The logarithm of the fg-bg stereo likelihood ratio, $\log(r_s)$. h) Thresholding on the joint likelihood ratio map $r_m r_s$ and removing hand area. i) Selecting the connected region in the object mask larger than a certain number of pixels, closest to the hand. j) Hole-filling, erosion and dilation of the one-region object mask.

articulated pose of the hand is represented.

The global position of the hand at time t is represented with the centroid of the hand mask relative to the centroid of the face mask. In Section V we experimentally compare a 2D position p_t^a in $\mathcal{I}_t^L$ (Figure 2a) with a 3D position q_t^a found by triangulation from $\mathcal{I}_t^L$ and $\mathcal{I}_t^R$ (Figure 2b).

Since articulated hand tracking is a difficult problem [12], [13], we use a holistic representation of hand shape without modeling individual fingers. The local articulated hand pose is represented using histograms of oriented gradients (HOG) [20], [21]. Gradient orientation $\mathbf{\Phi}_t \in [0, \pi)$ is computed from the segmented hand image $\mathbf{I}_t^a$ (Figure 2c) as $\mathbf{\Phi}_t = \arctan(|\frac{\partial \mathbf{I}_t^a}{\partial y}| / \frac{\partial \mathbf{I}_t^a}{\partial x})$ where x denotes downward (vertical) direction and y rightward (horizontal) direction in the image. The gradient orientation image $\mathbf{\Phi}_t$ is divided into $L \times L$ equal partitions. A histogram with B bins is computed from each partition. Figure 2c shows histograms overlayed on $\mathbf{I}_t^a$.

Earlier work [8] has shown that it is possible to reconstruct the articulated hand pose during a grasping action from the global 3D position of the hand and a HOG representation of local hand shape.

The hand pose at time t is represented by the vector x_t^a which is the concatenation of the 2D position p_t^a or 3D position q_t^a, and all histograms in the HOG. The performance of the classifier is quite insensitive to choices of $B \in [3, 8]$ and $L \in [2, 8]$; in our experiments in Section V we use $B = 4$ and $L = 4$. Before concatenation, p_t^a or q_t^a is normalized so that the standard deviations of the dimensions of x_t^a originating from p_t^a or q_t^a have the same standard deviation in the training set (Section V) as the remaining dimensions. The sequence of poses over the sequence is $\mathbf{x}^a = \{x_t^a\}, t = 1, \ldots, T$.

B. Object Features

The objects considered in this application are all "graspable", i.e. more or less rigid. For example, a cup is graspable, but water is not. Shape can therefore be expected to be a good object class descriptor, while local descriptors are unsuitable for our purposes.

Contrary to in many other object recognition applications, there is no search for object location involved. For manipulated objects, the position of the hand grasping the object gives an indication of the expected object position and motion, and the recognition problem becomes one of segmenting the object from the background using that information [22], and classifying the found image region. However, there might be deviations in position and orientation of the object within this region, as well as deviations in size, shape and color of different instances within each object class. The descriptor must therefore be insensitive to these intra-class variations, but still capture inter-class variations.

Given the hand positions p_t^L and p_t^R in the left and right images $\mathcal{I}_t^L$ and $\mathcal{I}_t^R$ at time t, and the hand position p_{t-1}^L in the left image $\mathcal{I}_{t-1}^L$ at time $t-1$, the object in the hand can be detected and segmented from the background in $\mathcal{I}_t^L$.

Note that we do not attempt to segment the whole scene into objects and background. Rather, objects are segmented as they are manipulated and thus move in a predictable manner.

Images are compared in terms of phase, i.e. gradient direction $\mathbf{\Psi}_t = \arctan(\frac{\partial \mathcal{I}}{\partial y}/\frac{\partial \mathcal{I}}{\partial x})$. This cue represents shape and texture rather than absolute pixel value; for example, a black object does not look more different from a light background than a white object with the same shape would.

The following assumptions are made:

1) The object is in the vincinity of the hand when manipulated. This prior information is exploited by only segmenting a window around the hand (Figure 3a).

2) The object is translated with the hand, with no in-plane or out-of-plane rotations. The likelihood of a pixel $p_t^L + \delta$ being foreground (Figure 3b) is therefore

$$P_m^{\text{fg}}(p_t^L+\delta) = G(\mathbf{\Psi}_t^L(p_t^L+\delta) - \mathbf{\Psi}_{t-1}^L(p_{t-1}^L+\delta); 0, \sigma_m) \quad (1)$$

where $G(\cdot; 0, \sigma_m)$ is a Gaussian with mean 0 and standard deviation σ_m. In our experiments, σ_m is equal to the width of the hand. Our backgound hypothesis is that the rest of the scene near the hand is static. This gives a background probability (Figure 3c) for each pixel $p_t^L + \delta$

$$P_t^{\text{bg}}(p_t^L+\delta) = G(\mathbf{\Psi}_t^L(p_t^L+\delta) - \mathbf{\Psi}_{t-1}^L(p_t^L+\delta); 0, \sigma_m)\ . \quad (2)$$

Object areas in the image thus tend to have high likelihood ratio $r_m = P_m^{\text{fg}}/P_m^{\text{bg}}$ while background has low ratio (Figure 3d). This assumption is valid also for slow object rotations, since the images are blurred (increasing image scale) before the phase computation, making the phase less noisy.

3) The object is at the same depth as the hand. In analog to the temporal assumption, the likelihood of a pixel $p_t^L + \delta$ being foreground (Figure 3e) is

$$P_s^{\text{fg}}(p_t^L + \delta) = G(\mathbf{\Psi}_t^L(p_t^L + \delta) - \mathbf{\Psi}_t^R(p_t^R + \delta); 0, \sigma_s) \quad (3)$$

i.e. maximal when the disparity at $p_t^L + \delta$ is identical to the hand translation $p_t^L - p_t^R$. In our experiments, $\sigma_s = \sigma_m$. The background hypothesis is that the pixel is at any depth (Figure 3f),

$$P_s^{\text{bg}}(p_t^L + \delta) = \frac{1}{N}\sum_{n=1}^{N} G(\mathbf{\Psi}_t^L(p_t^L + \delta) - \mathbf{\Psi}_t^R(p_t^L + [0, \rho] + \delta); 0, \sigma_s) \quad (4)$$

assuming no rotation between left and right camera coordinate systems. ρ is a random number, and $N = 100$ in our experiments. As with the temporal cue, object areas in the image tend to have high likelihood ratio $r_s = P_s^{\text{fg}}/P_s^{\text{bg}}$ while background has low ratio (Figure 3g). This assumption is valid if the depth changes over the object is much smaller than the object-camera distance.

4) The object has an outline with a low 2^{nd} order moment, i.e. few pointy features. When thresholding the joint likelihood ratio image $r_m r_s$ (assuming that the temporal and stereo cues are independent given the scene), a noisy object mask is obtained (Figure 3h). The threshold is close to 0, in our experiments $\exp(0.01)$. Assuming that the object is continuous, the connected foreground region larger than a certain number of pixels (a tenth of the total number of foreground pixels), closest to the hand is selected (Figure 3i). Furthermore, assuming that the object is "roundish" (with low 2^{nd} order moment) and without holes, the mask is processed using hole-filling, eroding and dilating morphological operations (Figure 3j).

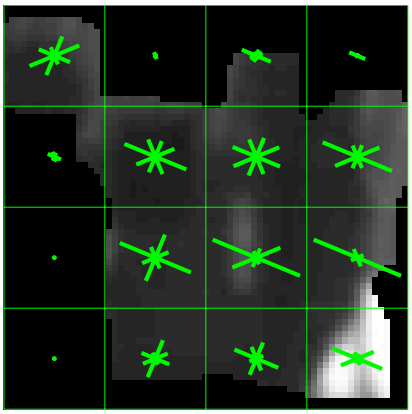


Fig. 4. Features used for object classification: Object image $\mathbf{I}_t^o$, HOG superimposed.

When the object is segmented from the background, the same HOG representation as for the hand shape is used. This description captures the shape of the object, with a certain insensitivity to absolute greylevels and small displacements of object parts (Figure 4). Note that this representation is not invariant to e.g. in-plane rotations; this is deliberate, since global orientation is indicative of object class in our application. The object at time t is represented by x_t^o, the concatenation of all histograms in the HOG.

While we here do not evaluate the object recognition performance in itself, HOG:s have previously [21], [5] been found to be robust to intra-class variability in shape, orientation, position, rotation and lighting conditions, while maintaining a good inter-class discriminability.

Another factor specific to this object recognition application is that the data consist of not only one, but a sequence of object views. In most cases, parts of the object are also occluded by the human hand grasping it. The change in orientation of the object with respect to the camera during the sequence and the occlusion from the hand are descriptive of the object, since they reflect the way this object class is used by the human; they can be termed "typical view sequences" and "typical occlusions". Thus, the classifier should take the whole sequence of object views into account. Each measurement is therefore described by a sequence of descriptors $\mathbf{x}^o = \{x_t^o\}, t = 1, \ldots, T$.

C. Correlation Between Action and Object Features

The classifier described in Section IV models explicit semantic dependencies between manipulation actions and the manipulated objects. However, there are also dependencies on the feature level.

The shape of the hand encoded in $\mathbf{x}^a$ gives cues about the object as well, since humans grasp different types of

objects differently, due to object function, shape, weight and surface properties. Similarly, the view change in $\mathbf{x}^o$ over the course of the sequence is correlated with the type of action performed with the object. This representation of the correlation between manipulation actions and manipulated objects is *implicit* and difficult to model accurately, but should be taken into account when modeling the simultaneous action-object recognition. The implications of the explicit and implicit object-action dependence are further investigated in our previous experiments in [5].

D. Temporal Segmentation of Action

A non-trivial but often overlooked problem in action recognition is temporal segmentation. Here, manipulation actions are defined as beginning with a pick-up event and ending with a put-down event, and temporal segmentation is possible if these events can be detected. For this, we use the fact that the object segmentation results in an empty mask if there is no large enough area of high likelihood ratio (Figure 3i). In the case of an empty mask, the frame is classified as "no object in hand", otherwise as "object in hand".

An example of a temporal segmentation is shown in Figure 8. To segment a sequence, each frame is classified as "no object in hand" or "object in hand". An eroding and a dilating operation are then performed over time, to remove spurious positive or negative classifications. Thereafter, all pick-up events (transfers from "no object in hand" to "object in hand") and put-down events (transfers from "object in hand" to "no object in hand") are detected. Periods with "object in hand" shorter than 1 sec are considered as noise and removed. The remaining periods are considered as primitive OACs to be classified.

The performance of the automatic temporal segmentation is evaluated in Section V.

IV. Discriminative Classification using CRF:s

Since we can expect complex dependencies within our action data $\mathbf{x}^a$ and object data $\mathbf{x}^o$ over time, a discriminative classifier which does not model the data generation process is preferable over a generative sequential classifier like a hidden Markov model (HMM) [23]. We thus employ conditional random fields (CRF:s) [24] which are undirected graphical models that represent a set of state variables $\mathbf{y}$, distributed according to a graph $\mathcal{G}$, and conditioned on a set of measurements $\mathbf{x}$. Let $C = \{\{\mathbf{y}_c, \mathbf{x}_c\}\}$ be the set of cliques in $\mathcal{G}$. Then,

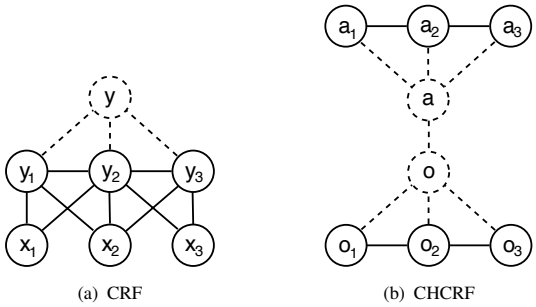
$$P(\mathbf{y}|\mathbf{x};\theta) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Phi(\mathbf{y}_c, \mathbf{x}_c; \theta_c) \tag{5}$$

where Φ is a potential function parameterized by θ as

$$\Phi(\mathbf{y}_c, \mathbf{x}_c; \theta_c) = e^{\sum_k \theta_{c,k} f_k(\mathbf{y}_c, \mathbf{x}_c)} \tag{6}$$

and $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in C} \Phi(\mathbf{y}_c, \mathbf{x}_c; \theta_c)$ is a normalizing factor. The feature functions $\{f_k\}$ are given, and training the CRF means setting the weights θ using belief propagation [24].

For linear-chain data (for example a sequence of object or action features and labels), $\mathbf{y} = \{y_t\}$ and $\mathbf{x} = \{x_t\}$, $t = 1, \ldots, T$ as shown in Figure 5a. This means that the cliques are the edges of the model, which gives



(a) CRF (b) CHCRF

Fig. 5. Different CRF structures used for action-object recognition with pre-segmented data. Dotted edges and nodes indicate that the weights θ associated with these are constrained during the training of the CRF. a) Linear-chain CRF [24]. b) Connected hierarchic CRF:s [5], data layer not shown.

$$P(\mathbf{y}|\mathbf{x};\theta) = \frac{1}{Z(\mathbf{x})} \prod_{t=2}^{T} \Phi(y_{t-1}, y_t, \mathbf{x}; \theta_t) \tag{7}$$

with a potential function

$$\Phi(y_{t-1}, y_t, \mathbf{x}; \theta_t) = e^{\sum_k \theta_{t,k} f_k(y_{t-1}, y_t, \mathbf{x})}. \tag{8}$$

Each state y_t can depend on the whole observation sequence $\mathbf{x}$ – or any subpart of it, e.g. the sequence $\{x_{t-\mathcal{C}}, \ldots, x_{t+\mathcal{C}}\}$, $\mathcal{C}$ being the *connectivity* of the model.

A CRF returns an individual label for each time-step, which means that the time-sequential data $\mathbf{x}$ to be classified do not have to be segmented prior to classification. However, if a segmentation is readily available, as in our case, the robustness of the classification is increased by assuming that for all labels within the segment, $y_t = y \, \forall \, t \in [1, T]$. With pre-segmentation, the model thus becomes $P(y|\mathbf{x};\theta) \equiv P(\mathbf{y} = [y, \ldots, y]|\mathbf{x};\theta)$ (dotted layer in Figure 5a).

In earlier work [5] we introduced a structure termed connected hierarchic CRF:s (CHCRF). As shown in Figure 5b, the assumption is that the overall action label a is correlated with the overall object label o, not the labels a_t and o_t of any particular frame. As indicated by the dotted edges and nodes, the probability of action a is approximated as $P(a|\mathbf{x};\theta) \equiv P(\mathbf{a} = [a, \ldots, a]|\mathbf{x};\theta)$ and the probability of object o as $P(o|\mathbf{x};\theta) \equiv P(\mathbf{o} = [o, \ldots, o]|\mathbf{x};\theta)$. The probability of a and o conditioned on the data $\mathbf{x}$ is

$$\begin{aligned} P(a, o|\mathbf{x};\theta) = & \frac{1}{Z(\mathbf{x})} \Phi(a, o, \mathbf{x}; \theta^{ao}) \\ & \prod_{t=2}^{T} \delta(a_{t-1} - a_t) \Phi(a_{t-1}, a_t, \mathbf{x}; \theta_t^a) \\ & \prod_{t=2}^{T} \delta(o_{t-1} - o_t) \Phi(o_{t-1}, o_t, \mathbf{x}; \theta_t^o) \\ = & \frac{\Phi(a, o, \mathbf{x}; \theta^{ao})}{\sum_{a,o} \Phi(a, o, \mathbf{x}; \theta^{ao})} P(a|\mathbf{x};\theta^a) P(o|\mathbf{x};\theta^o) \end{aligned} \tag{9}$$

where $\delta(\cdot)$ are Dirac functions ensuring that $a_{t-1} = a_t = a$ and $o_{t-1} = o_t = o$. To make the training more efficient,

(a) Instance of "look through binoculars"

(b) Instance of "drink from cup"

(c) Instance of "pour from pitcher"

Fig. 6. The three classes of the OAC dataset [5], performed by 10 different persons. Green = face mask, blue = hand mask, red = object mask.

the data dependency in the first term is omitted (indicated by the middlemost edge being dotted), becoming $\mathcal{K}(a,o) = \frac{\Phi(a,o;\theta^{ao})}{\sum_{a,o}\Phi(a,o;\theta^{ao})}$, the *co-occurrence rate* of action label a and object label o in the training data. The individual probabilities $P(a|\mathbf{x};\theta^a)$ and $P(o|\mathbf{x};\theta^o)$ can of course be estimated from any classification method that returns probability estimates. The parameters of the action and object classifiers are learned separately.

in a scenario with several OACs following each other, the object history can be taken into account. Every time an OAC i is detected in a certain scene, its classification $P_i^{\text{hist}} = P(a,o|\mathbf{x};\theta)$ and the final hand position p_i^{hist} in the image is stored. The underlying assumption is that an object only moves when manipulated by the human. When classifying a new OAC, the history $i = 1,\ldots,I$ is used as prior information:

$$P(a,o|\mathbf{x},\mathbf{P}^{\text{hist}},\mathbf{p}^{\text{hist}};\theta) = P(a,o|\mathbf{x};\theta)\frac{1}{I+1}\left(\frac{1}{\delta_0^{\text{hist}}}P_0^{\text{hist}} + \sum_{i=1}^{I}\frac{1}{p_i^{\text{hist}} - p_1^a}P_i^{\text{hist}}\right) \quad (10)$$

where $i = 0$ corresponds to the hypothesis that the object has not been manipulated before; δ_0^{hist} is a default distance, and P_0^{hist} is a uniform density.

V. Experiments

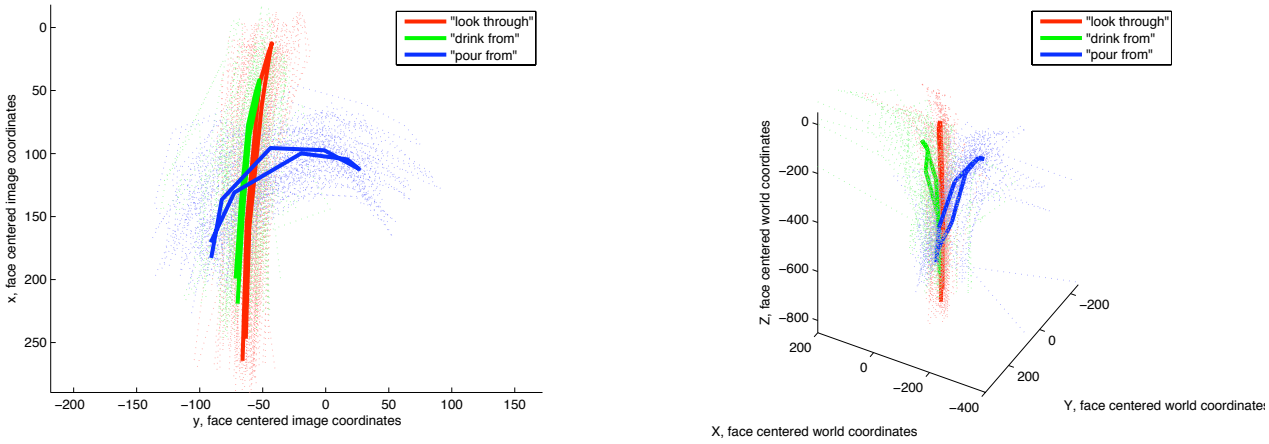
For training and testing of the object-action CHCRF, the OAC dataset was used [5]. The dataset consists of 50 temporally segmented instances each of three different action-object combinations; "look through binoculars", "drink from cup", and "pour from pitcher". The actions were performed by 7 men and 3 women, 5 times each. The classes were selected so that the object and action data are complementary: two of the actions, "look through" and "drink from" are similar, while "cup" and "pitcher" are similar. Three examples are shown in Figure 6.

In addition, a longer sequence was collected with different combinations of the three actions and objects in the OAC dataset, performed by one of the individuals in the OAC dataset. Ground truth temporal segmentation points indicating pick-up and put-down events were added manually, to enable testing of the automatic temporal segmentation. The three first OAC are correct in the sense that they are present in the training set, which means that they have a non-zero entry in the co-occurrence matrix $\mathcal{K}$. Of the last two previously unseen OACs, the first one, "drink from pitcher", is more plausible than the second one, "pour from binoculars", because a pitcher is similar to a cup both in terms of functionality and appearance than a pair of binoculars to a pitcher. Six frames are shown in Figure 8.

TABLE I

How important is the depth information to the action classification? % correct classification (median of 9 runs).

	2D	**3D**
Actions	95.3	95.3
Actions Baseline (only pos)	80	85.3



(a) Face centered 2D image hand position

(b) Face centered 3D world hand position

Fig. 7. Trajectories over hand position relative to face from the OAC dataset. a) 2D image hand position relative to face. b) 3D world hand position relative to face.

The first contribution of this paper compared to [5], modeling hand position in 3D, is first evaluated against the 2D representation. When testing with instances in the OAC dataset, training and testing was performed in a jackknife manner, where the 15 sequences of one individual at a time was used as a test data, the 135 sequences of the other 9 individuals as training set.

Figure 7 shows the 2D and 3D hand trajectories (dotted curves) in the OAC database. Although not used in training, the mean of each class is plotted (solid curves) for visibility. Apart from increasing the usability for learning from demonstration purposes, the estimated depth clearly increases the discriminability between the actions.

This is also supported by the classification results in Table I: Action classification using only 3D position $\mathbf{q}^a$ gives a slightly better result (85% correct classifications) than with only 2D position $\mathbf{p}^a$ (80% correct classifications). The difference is however diminished when adding the shape information (95.3% correct classifications). Obviously, the improvement from adding depth depends on the quality of the depth estimation; Figure 7b shows that the depth estimation from the center of gravities of segmented skin areas is quite noisy. A direction of future improvement is therefore to extract 3D human pose in a more accurate and robust way than skin color segmentation.

The main constribution of this paper, the spatial object segmentation and the temporal action segmentation was then evaluated with the longer sequence (Figure 8). For training, all sequences in the training set with the same individual as in the longer sequence were removed, leaving 135 sequences of the other 9 individuals for training of the CHCRF.

As shown in Figures 6 and 8, the object segmentation is successful in situations when the assumptions about object motion and position relative to the hand are not violated. However, if the object rotates (Figure 6c and a, third image) or differs too much from the hand in depth (Figure 6b, fourth image), the segmentation fails to find the correct object mask. A direction for future research is therefore to improve the object segmentation method so that it takes these common deviations from the segmentation assumptions into account.

The object segmentation is successful in the respect that it provides a correct temporal segmentation (Figure 8, middle bottom row); all the five actions are detected, with a mean error of 1.2 frames for pick-up and 3.4 frames for put-down compared to the ground truth.

Figure 8, top-most row, shows the CHCRF classification (Eq (9)) of each segmented OAC individually. Classification of the three first OACs is near-perfect, in concordance with the results in [5]. The results for the two last OACs show that the action classification is more accurated than the object classification, steering the OAC classification to a result that concords with the action class (”drink from cup”, ”pour from pitcher”). The object classification will improve with a better segmentation.

When the object history is taken into account, Eq (10), the object classification becomes more accurate. Figure 8, middle upper row, shows that the classifications of the conflicting two last OACs correspond to what can be expected: In ”drink from pitcher”, there are two options, most likely ”drink from cup” and less likely ”pour from pitcher”. This is reasonable, since the cup and the pitcher actually look similar, while the actions ”drink from” and ”pour from” look very different. In ”pour from binocular”, there are two approximately equally likeliy (or unlikely) options, ”pour from pitcher” and ”look through binoculars”. Consequently, the sum of unnormalized probabilities $Z(\mathbf{x})$, Eq (9), proportional to the overall probability of seeing this OAC, is much lower for ”pour from binoculars” (45000) than for ”drink from pitcher” (85000) and certainly lower than for the three previously seen OACs ($\mathcal{O}(10^7)$).

VI. Conclusions

A method for recognition of objects and manipulation actions from unsegmented video was presented. Our ap-

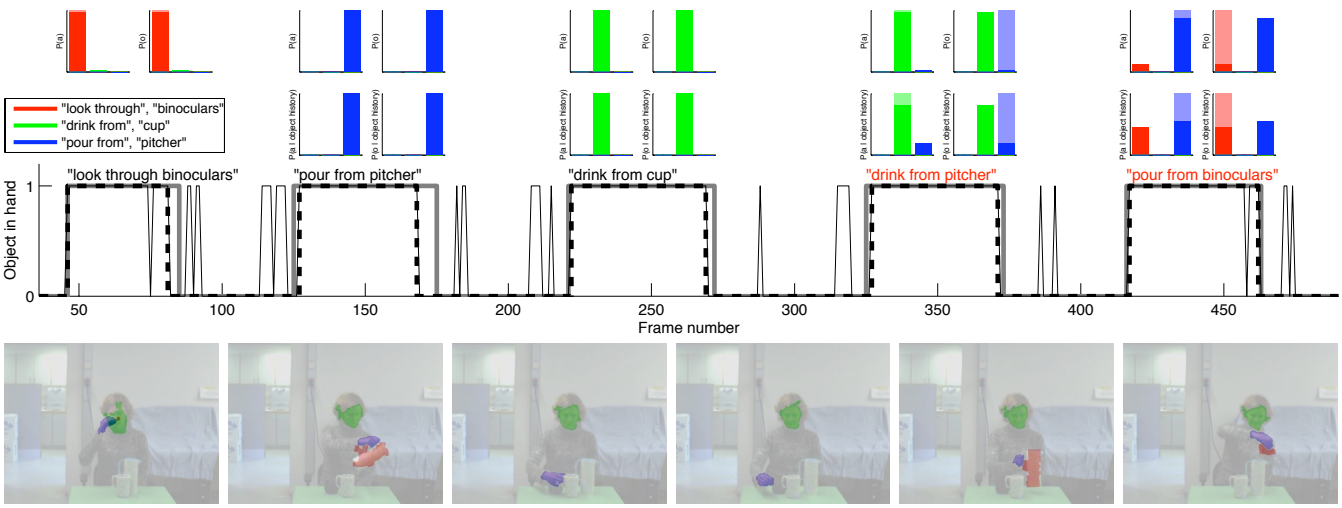


Fig. 8. Temporal segmentation and classification. Far bottom row) Spatial segmentation in frames 54, 135, 217, 291, 368 and 446 of the longer sequence. Middle bottom row) Temporal segmentation based on spatial object segmentation. Grey thick line is the ground truth, thin black line is the raw frame-by frame segmentations. Dotted thick black line is the segmentations processed as described in Section III-D. Far upper row) CHCRF classification of each of the 5 segmented OACs. The first three are present in the training set, the last two are previously unseen object-action combinations. Solid bars are classification results, semi-transparent bars are ground truth. Classes are color-coded according to the legend. Middle upper row) CHCRF classification where previous classifications of the same object have been taken into account.

plication of interest is robot learning of household tasks from human demonstration. A task can be described as a combination of primitive actions and objects drawn from a previously learned codebook. The intention is to train the codebook using the presented method, then recognizing primitive actions and objects in a new scene using the codebook. Future work include improving the object recognition to account for change in object state over time, improving the object segmentation method, as well as methods for representing composite tasks using the learned primitive actions and objects.

VII. Acknowledgments

This research has been supported by the EU through the project PACO-PLUS, FP6-2004-IST-4-27657, and by the Swedish Foundation for Strategic Research.

References

[1] A. Billard, S. Calinon, R. Dillman, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York, NY, USA: Springer-Verlag, 2008, ch. 59.

[2] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 37, pp. 286–298, 2001.

[3] J. J. Gibson, *The Ecological Approach to Visual Perception*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1979.

[4] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr, "Cognitive agents – a procedural perspective relying on the predictability of object-action-complexes (OACs)," *Robotics and Autonomous Systems*, in press.

[5] H. Kjellström, J. Romero, D. Martínez, and D. Kragic, "Simultaneous visual recognition of manipulation actions and manipulated objects," in *ECCV*, vol. 2, 2008, pp. 336–349.

[6] L. Y. Chang, N. S. Pollard, T. M. Mitchell, and E. P. Xing, "Feature selection for grasp recognition from optical markers," in *IROS*, 2007.

[7] K. Ogawara, J. Takamatsu, K. Hashimoto, and K. Ikeuchi, "Grasp recognition using a 3D articulated model and infrared images," in *IROS*, vol. 2, 2003, pp. 1590–1595.

[8] H. Kjellström, J. Romero, and D. Kragic, "Visual recognition of grasps for human-to-robot mapping," in *IROS*, 2008.

[9] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in computer vision-based human motion capture and analysis," *CVIU*, vol. 104, no. 2-3, pp. 90–126, 2006.

[10] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *PAMI*, vol. 19, no. 7, pp. 677–695, 1997.

[11] M. Yang, N. Ahuja, and M. Tabb, "Extraction of 2d motion trajectories and its application to hand gesture recognition," *PAMI*, vol. 24, no. 8, pp. 1061–1074, 2002.

[12] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," *PAMI*, vol. 28, no. 9, pp. 1372–1384, 2006.

[13] E. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, "Visual hand tracking using non-parametric belief propagation," in *IEEE Workshop on Generative Model Based Vision*, 2004.

[14] E. Rivlin, S. J. Dickinson, and A. Rosenfeld, "Recognition by functional parts," *CVIU*, vol. 62, no. 2, pp. 164–176, 1995.

[15] L. Stark and K. Bowyer, *Generic Object Recognition using Form and Function*. World Sci. Ser. Machine Perception and Artificial Intelligence; Vol. 10, 1996.

[16] R. Mann and A. Jepson, "Towards the computational perception of action," in *CVPR*, 1998.

[17] D. J. Moore, I. A. Essa, and M. H. Hayes, "Exploiting human actions and object context for recognition tasks," in *ICCV*, 1999.

[18] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, "A scalable approach to activity recognition based on object use," in *ICCV*, 2007.

[19] A. A. Argyros and M. I. A. Lourakis, "Real time tracking of multiple skin-colored objects with a possibly moving camera," in *ECCV*, vol. 3, 2004, pp. 368–379.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 2, 2005, pp. 886–893.

[21] K. Mikolajczyk, B. Leibe, and B. Schiele, "Local features for object class recognition," in *ICCV*, 2005, pp. 525–531.

[22] D. Zhang and G. Lu, "Segmentation of moving objects in image sequence: a review," *Circuits Systems Signal Processing*, vol. 20, no. 2, pp. 143–183, 2001.

[23] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[24] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

ROYAL INSTITUTE
OF TECHNOLOGY

School of Computer Science and Communication

Image Segmentation of Manipulation Actions with Level Sets

David Martínez, Javier Romero, Hedvig Kjellström, Danica Kragić

Technical Report

TRITA-CV 2008:2

David Martínez, Javier Romero, Hedvig Kjellström, Danica Kragić

Image Segmentation of Manipulation Actions with Level Sets

Report number: TRITA-CV 2008:2
Publication date: December 2008
E-mail of author: dvdmartinez@gmail.com, jrgn,hedvig,dani@kth.se

Reports can be ordered from:

School of Computer Science and Communication (CSC)
Royal Institute of Technology (KTH)
SE-100 44 Stockholm
SWEDEN

telefax: +46 8 790 09 30
http://www.kth.se/csc/

Image Segmentation of Manipulation Actions with Level Sets

David Martínez, Javier Romero, Hedvig Kjellström, Danica Kragić
Computational Vision and Active Perception Lab (CVAP)
Centre for Autonomous Systems (CAS)
School of Computer Science and Communication (CSC)
Royal Institute of Technology (KTH)
SE-100 44 Stockholm
SWEDEN

Abstract

This report summarizes the work done by the first author during the spring of 2008 at KTH, within the EU project PACO-PLUS, FP6-2004-IST-4-27657 (`www.paco-plus.org`).

Robots are usually taught by writing thousands of lines of code. In the future, a more natural way will be used, called Programming by Demonstration (PbD); somebody will show a robot how to do something just doing it; the robot will learn how to do that. In order to achieve this, the robotmust be able to get, from a video sequence, the movements of the person, and find out what they represent, for instance, a grasping. It will also need to find the object/s that the user is using to recognize them; in the last term, if the robot does not recognize the object, it could simply ask "what's that?", as humans do.

We propose a level set method that integrates different cues for the segmentation of moving objects inside a video sequence; those objects are manipulated by a person who is teaching a robot how to do something. We also obtain a segmentation of the person's hands and head in order to retrieve actions. Objects, hands and the head are tracked along the sequence to help the segmentation and deal with occlusions. During the sequence, new objects can appear in the scene.

1 Introduction

This report summarizes the work done by the first author during the spring of 2008 at KTH, within the EU project PACO-PLUS, FP6-2004-IST-4-27657 (`www.paco-plus.org`).

A method for segmentation of a scene containing a human and objects manipulated by the human is presented. The segmentation is intended as a pre-step to a method for jointly recognize actions and objects [1].

At the beginning of the video sequence the person is outside the range of vision of the robot, and then he begins entering the scene; after a few frames the user is already in front of the robot, and the segmentation of moving objects can start.

Each frame consist of 3 layers: the background, a layer with the person, and a layer with moving objects; the hands and the head also belong to the object's layer, as they can be treated as moving objects.

The key idea is that the level set method can be used to integrate different cues to segment an image, in our case to segment moving objects being manipulated by a user. Those cues alone do not need to be very reliable. As it is common in level set approaches, colour and texture can be used to improve the segmentation.

The report is outlined as follows. Section 2 discusses work related to the present method. In Section 3, our approach is described. Section 4 discusses the automatic vs manual learning of thesholds in our system. Section 6 goes deeper into details about the implementation, while Section 7 concludes the report.

2 Related Work

The segmentation of moving objects in a video sequence is a well studied vision problem. Foreground layer extraction [2,3] is a fast and reliable method, but it is not able to differenciate among moving objects. A more general approach is to find regions of coherent motion to split the sequence into layers [4]; if these methods rely too much upon motion, problems may arise with non-textured regions, fast moving objects, etc. It is interesting to integrate different cues besides motion; two recent frameworks for image segmentation are able to integrate common cues like intensity, colour, texture, motion or stereo. Graph cuts [5] and normalized cuts [6] consider an image as a graph whose nodes are the pixels; the segmentation is performed by minimizing an energy functional, and this minimization is done by finding minimum cuts of the graph. The other framework is called level set [7,8]; it is an elegant way of evolving one or more contours implicitly represented by a level set function.

3 Our Approach

As explained before, we use a level set method to integrate different cues in order to segment the hands and the head of the preson and the moving objects. While the skin model is the same for all the sequences, we can not expect to have the same background or a person wearing the same clothing. Thus, we need a training phase at the beginning of each sequence to learn the background and the person's model. We consider that at the beginning the robot can only see the background, not the person; when he comes into the scene, we can detect him with the background substraction method explained in subsection 3.5. We can then build a model for the person and another one for the background. After several frames, the segmentation phase will begin. In a real setup, the person could be the one who tells the robot when to begin.

3.1 Level Set Method

Let $I : \Omega \rightarrow \mathbb{R}^n, \Omega \subset \mathbb{R}^2$ be the image we want to segment, where n is the number of channels (3 for a RGB image). We want to segment the image into a set of t regions $\{R_i\}_{i=1}^t$ so that they cover the full image, $\cup_{i=1}^t R_i = \Omega$; an

additional restriction is that they must be pairwise disjoint to avoid ambiguities, $R_i \cap R_j = \emptyset, \forall i \neq j$.

We use a set of curves $\{\Gamma\}_{i=1}^{N}$ implicitly represented via level set functions $\phi : \Omega, \mathbb{R}^+ \rightarrow \mathbb{R}$ to divide the image into regions; the curve Γ_i is then represented by the zero level set of $\phi_i(x, y, t)$: $\Gamma_i = \{(x, y) \in \Omega | \phi_i(x, y, t) = 0\}$. The level set function evolves as time increases according to a specified energy functional. The regions inside and outside the curve can be distinguished by the sign of ϕ_i at each pixel of the image. ϕ must be a signed distance function to use the level set method.

The level set is used to minimize an energy functional E that depends on the image and ϕ. The evolution equation that drives the zero isocontour of ϕ is

$$\frac{\partial \phi}{\partial t} = -\frac{\partial E}{\partial \phi} \tag{1}$$

Following the active geodesic regions model [9], if we consider only 2 regions, the energy functional to minimize has this form:

$$E = -\int_{\Omega_1} \log(p_1(x))dx - \int_{\Omega_2} \log(p_2(x))dx + \alpha \int_{\Omega} |\nabla H(\phi)|dx \tag{2}$$

where $p_1(x)$ and $p_2(x)$ are the probability density functions for regions Ω_1 and Ω_2, respectively. In this model it is assumed that both regions have the same prior probability, and that pixels are independent.

The last part of equation 2 is called a regularization term that measures the length of the zero level set, and is used to smooth the segmentation; α is a parameter. $H(\phi)$ is the Heaviside function.

The integrals can later be extended to the whole domain Ω:

$$\begin{aligned} E = \quad & - \int_{\Omega} \log(p_1(x))H(\phi)dx \\ & - \int_{\Omega} \log(p_2(x))(1 - H(\phi))dx \\ & + \alpha \int_{\Omega} |\nabla H(\phi)|dx \end{aligned} \tag{3}$$

Finally, the evolution equation becomes

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left(\log(p_1(x)) - \log(p_2(x)) + \alpha div \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right) \tag{4}$$

As usual, we use smoothed approximations of the Heaviside and delta functions [7]:

$$H(\phi) = \begin{cases} 0 & \phi < -\epsilon \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi} \sin\left(\frac{\pi\phi}{\epsilon}\right) & -\epsilon \leq \phi \leq \epsilon \\ 1 & \epsilon < \phi \end{cases} \tag{5}$$

$$\delta(\phi) = H'(\phi) = \begin{cases} 0 & \phi < -\epsilon \\ \frac{1}{2\epsilon}(1 + \cos\left(\frac{\pi\phi}{\epsilon}\right) & -\epsilon \leq \phi \leq \epsilon \\ 0 & \epsilon < \phi \end{cases} \tag{6}$$

3.2 Multiple Regions

We want to segment each frame into 4 regions: one for the skin, one for the person, one for the moving objects and another one for the background. We will not always have all these regions, as occlusions may appear, the user might not be manipulating anything, etc. If we consider 1 level set per region, then equation 4 can be used for each level set; in this case we need two models for each region, the model of the region we want (i.e the skin model), and a model of what we do not want (i.e. the background). With this system of 4 independent evolution equations we cannot guarantee to obtain disjoint regions Ω.

Instead, we will use an approach that guarantees to obtain pairwise disjoint regions; moreover, the number of models we need is reduced. We still need one level set per region to segment, but in this case the model of that region competes with all the remaining ones [10]. The evolution equations become

$$\frac{\partial \phi_i}{\partial t} = \delta(\phi_i) \left(e_i(x) - \max_{j \neq i, H(\phi_j) > 0} e_j(x) + \frac{\alpha}{2} div \left(\frac{\nabla \phi_i}{|\nabla \phi_i|} \right) \right) \tag{7}$$

The resulting regions do not form a partition of the image because there are some pixels that are not assigned to any region due to numerical inaccuracies; those pixels are placed on the limits of the regions, and thus can be ignored.

3.3 Skin

For the skin we use a GMM with 4 Gaussians. The parameters for the mixture distribution are found with the Expectation-Maximization algorithm. The skin model does not depend on the sequence, so it can be obtained in advance and be used for all the sequences.

3.4 Person

Several frames at the beginning of the sequence are used to build the person's model. During the first frames we consider that the person is out of the video, and then he enters; we also consider that in those frames there is nothing else moving. Those moving regions of the video will belong to the person; we use Stauffer's approach [11] to detect the moving regions; we explain it in detail in subsection 3.5. Once we know for each frame which pixels belong to the person, we can construct a Gaussian Mixture Model of the person and another one for the background. The number of Gaussians is chosen with the Bayesian Information Criterion [12]:

$$BIC = -2 \ln(L(x_1, \ldots, x_N)) + k \ln(N) \tag{8}$$

where $x_1, \ldots, x_N$ is the set of N pixels used to obtain the model, L is the likelihood given by the model for our dataset and k is the number of parameters, which for a d-dimensional GMM with t Gaussians is $k = t(d+1)(d+2)/2$. Several GMM with different t are constructed, and the preferred is the one with minimum BIC.

To improve the model, one can use data from several frames; instead of keeping all the points from all the frames, which is very slow and needs a lot of memory, one can construct a GMM per frame, and then use statistical criteria to mix, update or remove Gaussians from all the models until we get a global one.

3.5 Background

In order to detect moving objects, we need something that can deal with very long movements; currently we implement Stauffer's and Grimson's approach [11] for background subtraction; in this approach each pixel is modelled with a GMM with K Gaussians (3 in our case); thus, the probability of observing a pixel value X at position (x, y) is

$$P(X) = \sum_{i=1}^{K} w_{(x,y),i} \eta(X, \mu_{(x,y),i}, \Sigma_{(x,y),i}) \tag{9}$$

where $\eta(X, \mu_{(x,y),i}, \Sigma_{(x,y),i})$ is the Gaussian pdf. To check if a pixel is moving or not we check it against all the Gaussians for that pixel position. If there exist a Gaussian distribution that matches well the pixel X (if $|X - \mu| < 2.5\sigma$), it is considered as background.

This approach is dynamic, and after each frame an update step is done to adapt to illumination changes, shadows, etc. If there exist a good match for pixel p, whose value is X_t, the following formulas can be used:

$$\begin{aligned}
&w_{p,k,t} = (1 - \alpha) w_{p,k,t-1} + \alpha M_{p,k,t} \forall k = 1, \ldots, K \\
&\mu_{p,best_k,t} = (1 - \rho)\mu_{p,k,t-1} + \rho X_t \\
&\sigma^2_{p,best_k,t} = (1 - \rho)\sigma^2_{p,k,t-1} + \rho(X_t - \mu_{p,k,t})^T (X_t - \mu_{p,k,t})
\end{aligned} \tag{10}$$

where α is the learning rate (we use $alpha = 0.05$), and ρ is a learning factor for adapting current distributions; it can be $\rho = \alpha\eta(X, \mu_{(x,y),i}, \sum_{(x,y),i})$ or, to speed up the update, it can be set to a constant; we just use $\rho = \alpha$. $M_{p,k,t}$ is 1 for the best matched Gaussian and 0 for the rest. Notice that, while all the weights are updated at each time, only the mean and variance of the Gaussian that best matched the pixel are updated; the other Gaussians are kept the same. Also, the weights must be renormalized so that $\sum_{i=1}^{k} w_{p,k,t} = 1, \forall p$.

If there is no good match for the pixel, the worst Gaussian is replaced by a new one whose mean is the pixel value, and with a high variance and low weight.

Stauffer's approach only tells us which pixels do (not) belong to the background, but not their motion vectors. When the user is moving the model will detect it as motion, but when he stops moving, after several frames parts of the person will be considered as background, something that we do not want; for that reason, we avoid updating pixels that belong to the person, so they are always detected as motion, even if the person is still; it is a way of improving the background model in the long term. The same can be done with the moving objects once they have been detected and tracked.

3.6 Object

So far, we have models for the skin, for the person and for the background, but, as the object has not yet been recognized, we have no model for the object. We know that the object will be moving (we only recognize objects when they move), and that the other models will give low likelihoods. We can use one

threshold per model to decide when a pixel is well matched with that model. Then, pixels that are not matched to any of our three models, and that are moving, will belong to an unrecognised object.

From here, there are options: 1) Get the object's mask, find the blobs and recognize objects with the object-action classifier [1], keeping in memory their positions after they have been put down by the human.

2) Once a new object is detected, add a new level set for that object, building a new model for each one; the general level set for objects will be used only in case there is a new object.

We are using approach 1, as it is faster and easier to implement, although approach 2 is more elegant.

3.7 Tracking

We can track objects even when they are not moving to know their position. When they are moving it can be useful because we can predict the position at the next frame. When they are not moving they will be integrated after several frames with the background model unless we explicitly avoid it, as explained before.

The initial level set curves for a frame $i+1$ are the final curves from the previous frame i; this speeds up the convergence of the method, as long as the moving objects for frame $i+1$ are not too far from the position at the previous frame. In that case, if there is no curve close to the moving object, the convergence may be very slow, or we could even reach the maximum number of iterations without a proper segmentation. In this case the tracking is useful because it tells us if we have this problem. Possible solutions could be to use a standard initialization, to artificially translate the curve to the new predicted position, or to keep the curve where it is and add small curves around the predicted new position. The implicit representation of the curves allows us to do all the things explained before; all the choices should be analysed, although the most natural seems to be the third one.

The hands and the head are also tracked (actually they are also considered as objects). In the case that several moving objects appear (we consider that at most 2 objects could be moving, one per hand), the position of the hands and the moving objects and the tracking mechanism will help us decide which ones are actually being manipulated by the user.

This tracking is also useful when the person is using something without moving it; for instance, if the action is pouring water onto a glass, the user will grasp the glass, but he will not move it. If we have tracked before the glass, then we can find out that it is being used.

An $\alpha - \beta$ tracker is used. With a high frame rate it should work really good, and it should only fail with sudden and sharp movements. To do tracking, we assume that objects, hands and head can be modelled as ellipses, and then we use Argyros tracker [13].

4 Thresholding

We have a model for the skin, for the person and for the background, but we lack of such a model for the objects, as we do not know in advance what kind

of objects we will have. To cope with this, thresholding is used to decide when a pixel does not match those models well. Pixels that do not match any model and that are moving will be recognized as objects. The skin threshold can be set in advance, while the person and background are found in the training phase.

5 Implementation Details

One common way of speeding up the segmentation is the use of a multiscale approach; the level sets are first computed on a downsampled image, and the obtained segmentation is used to initalize the algorithm at a coarser scale. We use 2 scales, 60x80 pixels and 240x320.

High order methods, as proposed in [7], can be used for the spatial derivatives; for the temporal derivatives we can use a Runge-Kutta method. But, instead, we have used simple central differencing and the Euler method, as higher order methods do not increase the accuracy of the segmentation.

To detect convergence and stop the evolution, we get the length of the contours after each iteration, and when their change is not significant (below a threshold) during several frames, the process is stopped. Also, we guarantee that the number of iterations is in the range $[IT_{min}, IT_{max}]$, even if the previous condition is satisfied before IT_{min}. This convergence criteria is explained in [14]. These parameters have been set manually.

ϕ must be always a signed distance function [7], but after several iterations this does not happen. To solve this, a common choice is the use of a reinitialization function that modifies ϕ without moving the curve so it is again a signed distance function. Instead, we add another term to the energy functional, as explained in [15], that forces ϕ to be always close to a signed distance function:

$$E_{final}(\phi) = \mu P(\phi) + E(\phi), \text{with } P(\phi) = \int_{\Omega} \frac{1}{2}(|\nabla\phi| - 1)^2 dx \qquad (11)$$

where $\mu > 0$ is a parameter that penalizes the deviation of ϕ from a signed distance function.

6 Conclusions

This report summarized the work done by the first author during the spring of 2008 at KTH, within the EU project PACO-PLUS, FP6-2004-IST-4-27657 (`www.paco-plus.org`).

A method for segmentation of a scene containing a human and objects manipulated by the human was presented. The segmentation was intended as a pre-step to a method for jointly recognize actions and objects [1].

This work will be continued by Master students within the PACO-PLUS project during the winter and spring of 2009.

References

[1] H. Kjellström, J. Romero, D. Martínez, and D. Kragic, "Simultaneous visual recognition of manipulation actions and manipulated objects," in *European Conference on Computer Vision*, vol. 2, pp. 336–349, 2008.

[2] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Bilayer segmentation of binocular stereo video," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005.

[3] J. Sun, W. Zhang, X. Tang, and H. Shum, "Background cut," in *European Conference on Computer Vision*, pp. 628–641, 2006.

[4] J. Wang and E. Adelson, "Representing moving images with layers," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 625–638, 1994.

[5] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[7] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*. Springer-Verlag, New York, 2002.

[8] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 195–215, 2007.

[9] N. Paragios and R. Deriche, "Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision," *Journal of Visual Communication and Image Representation*, vol. 13, no. 1/2, pp. 249–268, 2002.

[10] T. Brox and J. Weickert, "Level set based image segmentation with multiple regions," *Pattern Recognition*, vol. 3175, pp. 415–423, 2004.

[11] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[12] S. Calinon, F. Guenter, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," in *IEEE International Conference on Robotics and Automation*, 2006.

[13] A. Argyros and M. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera," in *European Conference on Computer Vision*, vol. 3, pp. 368–379, 2004.

[14] K. Chaudhury and K. Ramakrishnan, "Stability and convergence of the level set method in computer vision," *Pattern Recognition Letters*, vol. 28, no. 7, pp. 884–893, 2007.

[15] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without reinitialization: A new variational formulation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 430–436, 2005.

Markerless Human-to-Robot Grasp Mapping based on a Single View

Javier Romero Hedvig Kjellström Danica Kragić
Computational Vision and Active Perception Lab
Centre for Autonomous Systems
CSC, KTH, SE-100 44 Stockholm, Sweden
jrgn,hedvig,dani@kth.se

***Abstract*— The paper presents a system for robot grasp generation in an imitation based framework. The grasp observation is based on visual classification of human grasps using only a single image without any markers. Grasp are generated and evaluated on a three fingered Barrett hand and a five fingered Karlsruhe hand. The orientation of the hand relative to the objects is taken into account during the execution of the grasp. The contributions of the paper are the markerless grasp classification system, the analysis of grasp mapping and the use of virtual fingers approach for the evaluation of mapping quality. Experimental evaluation is performed both in simulation and on the real robot where tactile feedback is used to guide the grasp execution.**

I. Introduction

During the last decade, detection, representation and mapping of human activities to robots have been important areas of research in the field of robot learning through imitation and demonstration, [1]. There are examples of systems performing imitation of the arm motion [2] or, more generally, the upper-body [3], [4]. The systems used to measure the motion of the human consider mainly magnetic trackers although there have also been examples of vision based tracking systems [4]. For robots that are intended to perform human-like tasks in natural environments, grasping and manipulating objects is a necessary skill. In humans, during object grasp, there is a coordinated activation of distal muscles that supports to shape the hand in relation to the physical properties of the object. The human hand has extreme flexibility as a manipulator but there is still very little known of the muscular activation patterns that allow objects of different sizes and shapes to be grasped. In addition, complex movements of the hand are challenging to measure, model and imitate.

Programming robots for manipulating various objects in natural environments is difficult and equipping robots with the ability to learn from observing a human may reduce the cost of explicitly programming the new grasping behaviors. For objects of same/similar geometry there may potentially be many different grasps, dependent on their function. It is also important that the robot can learn from humans without any need of specialized sensors or markers, performing the grasping action as naturally as possible. In our previous work, [6] we have presented a vision based grasp recognition system that recognizes grasps from a single camera image independent of the viewing angle. The system is able to classify grasps according to a predefined set of grasp classes and also estimates the orientation of the grasp using

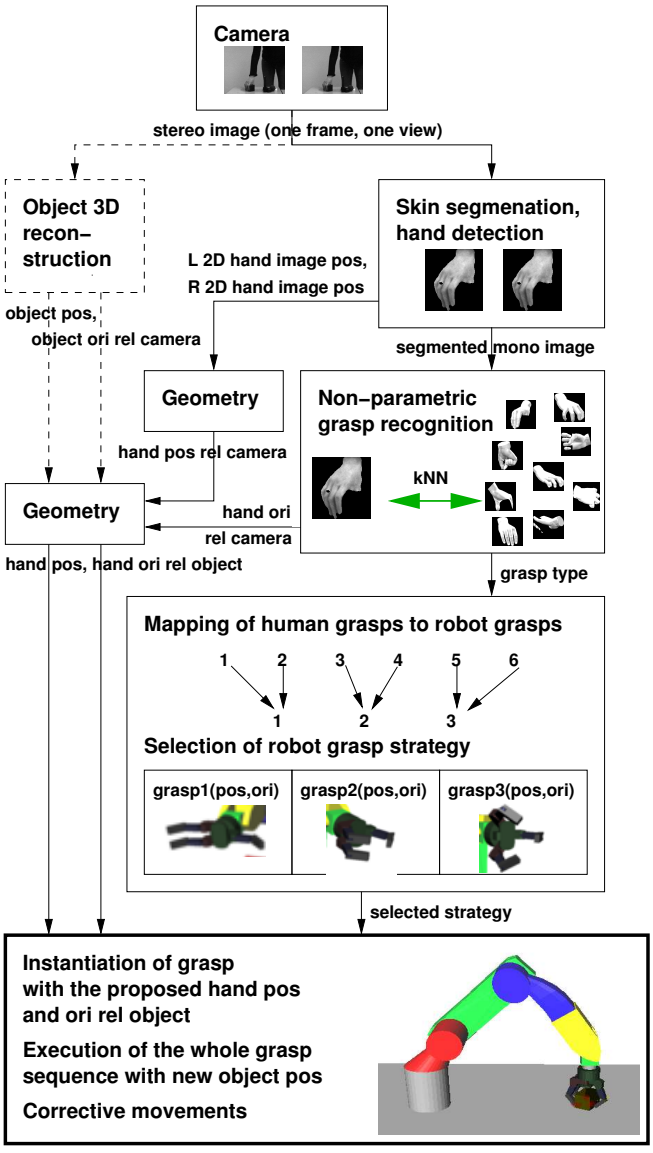
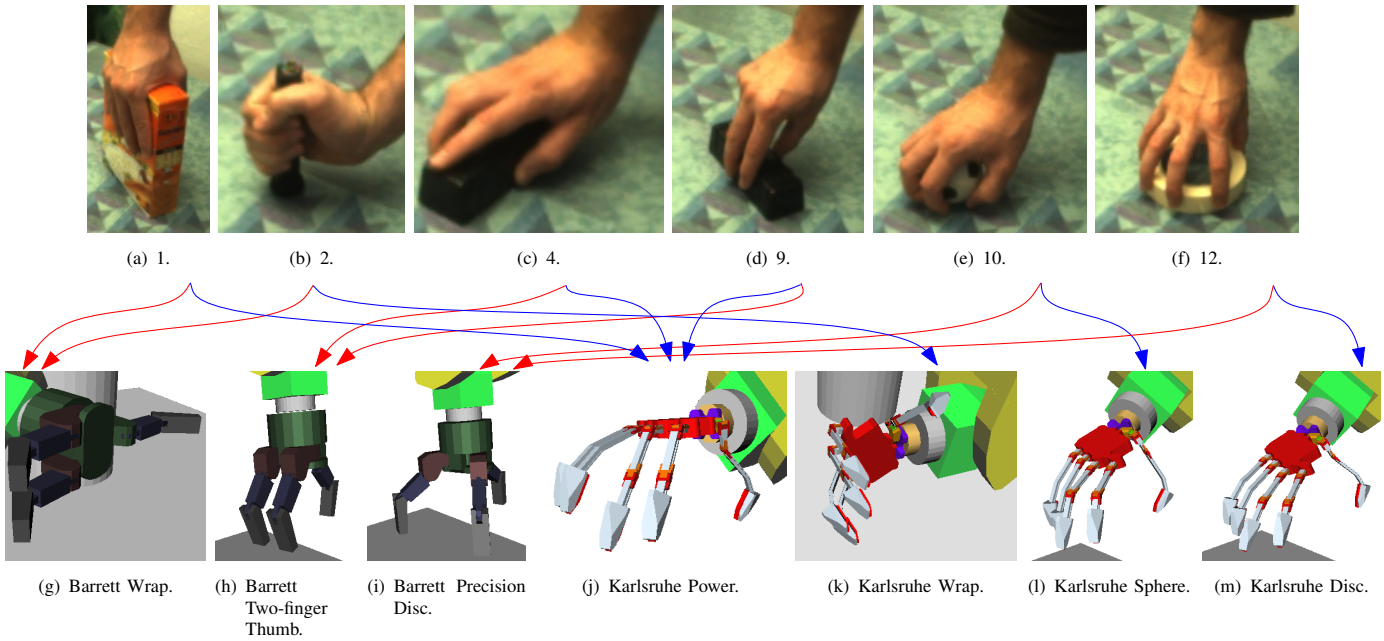


Fig. 1. Human grasps are recognized and mapped to a robot. From one video frame, the hand is localized and segmented from the background. The hand orientation relative to the camera, and type of grasp is recognized by a non-parametric classifier/regressor. The human grasp class is mapped to a corresponding robot grasp, and a predefined grasp strategy, *the whole approach-grasp-retreat sequence*, for that grasp is selected. The strategy is parameterized with the orientation and position of the hand relative to the object. Using this strategy, the robot performs the grasping action.



(a) 1. (b) 2. (c) 4. (d) 9. (e) 10. (f) 12.

(g) Barrett Wrap. (h) Barrett Two-finger Thumb. (i) Barrett Precision Disc. (j) Karlsruhe Power. (k) Karlsruhe Wrap. (l) Karlsruhe Sphere. (m) Karlsruhe Disc.

Fig. 2. The six grasps (numbered according to Cutkosky's grasp taxonomy [5]) considered in the classification, and the three grasps for a Barrett hand and Karlsruhe hand, with human-robot class mappings ((a,b)→(g),(c,d)→(h), (e,f)→(i)), (a,c,d)→(j), (b)→(k), (e)→(l), (f)→(m) shown. a) Large Diameter grasp, 1. b) Small Diameter grasp, 2. c) Abducted Thumb grasp, 4. d) Pinch grasp, 9. e) Power Sphere grasp, 10. f) Precision Disc grasp, 12. g) Barrett Wrap. h) Barrett Two-finger Thumb, i) Barrett Precision Disc, j) Karlsruhe Power, k) Karlsruhe Wrap, l) Karlsruhe Sphere, m) Karlsruhe Disc

regression. Since it is known that the velocity profile of the human hand movement reaches the minimum once the grasp is to be performed and just before the object is to be moved, this knowledge can be used to acquire an image used for classification and final mapping to a robot, see Figure 1.

In this paper, we present how the grasp classification is used to map grasps to a robot hand. While articulated 3D reconstruction of the hand is easy when using magnetic data or markers, 3D reconstruction of an unmarked hand from images is an extremely difficult problem due to the severe self-occlusion [7], [8], [9], [10]. Our method classifies grasp types and estimates their orientation from a single image and from any viewpoint, without building an explicit representation of the hand, similarly to [9], [11]. As outlined in the next section, most state of the art methods perform classification only from a single viewpoint or employ special sensing devices such as datagloves, magnetics sensors or different type of markers.

Grasp classification is here formalized as the problem of classifying a hand posture as one of six grasp classes, labeled according to Cutkosky's grasp taxonomy [5]. The classes are, as shown in Figure 2a-f, Large Diameter grasp, Small Diameter grasp, Abducted Thumb grasp, Pinch grasp, Power Sphere grasp and Precision Disc grasp. The mapping is then made to two different robotics hands, the three fingered Barrett hand and the five fingered Karlsruhe hand, see Figure 2g-i. The four degrees of freedom and three fingers of the Barrett hand allow us to define just three different preshapes. For the Karlsruhe hand, Abducted Thumb, Pinch and Large Diameter are mapped to the same preshape. We also note here that the distinction between the grasps is not made just in terms of the hand preshape, but also in terms of different strategies for approaching the objects; as pointed out by [12] the approach strategy for a precision grasp is different to the approach strategy for a power grasp. Apart from the classification and mapping, the important contribution of the paper is the use of *virtual fingers*, [13] for performing a quantitative analysis of grasp mapping.

In Section III we describe grasp mapping that also uses the tactile sensors on the robot hand to perform successful grasps. In Section IV we present the experimental evaluation in terms of similarity between contact positions and virtual fingers ([13], [12]) as well as the experiments with the real robot. Finally, Section V concludes the paper.

II. Related Work

According to [14], grasping in humans involve two well separated components: the approach component (involving the arm muscles) and the grasp component (involving the hand muscles). Despite that it has been showed that these systems are closely correlated, [15] research in robotics related to the imitation focuses mainly on one of the two components.

In the field of robotics, there are examples of systems that perform grasp execution based on the object geometrical properties, [16], [17], [18], [19], [20], [21] human observation or combination of both, [22]. [16] and [17] create a database of optimal grasps for objects whose 3D models are known which is then used online assuming object recognition. In [18], the authors use an offline learning approach to find a grasping point on an object for two-fingered grasps. In [19], complex objects are modeled with

a set of boxes to allow for grasp planning. Given the 3D model of an object, [21] use a database to extract a hand pose that fits the shape of the object, based on the normal to the object and hand in different random points. Finally, [23] classify objects based on their affordances (categories like "sidewall-graspable"), so the classification itself determines how to grasp the object.

For the above examples, once the training has been performed the systems perform well. However, it has been showed in [24] that appropriate usage that affects grasp types of objects requires information not related just to the objects' structure. For example, the way a human grasps a hammer is not the most natural or most stable for this object, but it is the best for the purpose a hammer is used. It is further argued that a longterm experience with handling objects comes into play and influences the interaction. Rather than reprogramming the robot system once the function of the object has to be taken into account, the approaches based on human demonstration provide the necessary experience in a natural way. In addition, such systems can be integrated with any of the above systems to provide a set of grasps for the objects stored in the database.

Classification of hand pose is most often used for gesture recognition, [9], [25], often characterized by low or no occlusion of the hands, and visually disparate set of hand poses. In our case, the grasped object usually occludes large parts of the grasping hand and different grasping poses may be very similar, [6]. Related approaches to grasp recognition [26], [27] first reconstruct the hand in 3D and use it for classification but the latter approach notes that the full 3D reconstruction is not needed to recognize grasp type.

Once the information about human grasp is available, it has to be mapped to the robot. This step is highly dependent on the dexterity of the robotic hand: in [28], mapping to a parallel jaw gripper has been done just by setting the gripper positions to the position of two fingertips. However, mapping to more complex robot hands is more complicated. In [13] the concept of virtual finger is introduced where one or more real fingers are acting in unison. [12] use this concept as an intermediate step in the mapping procedure. We use this work to evaluate the quality of our mapping strategy.

III. Grasp Classification and Mapping

The estimated grasp class as well as hand and object orientation and position are used to instantiate a robotic grasp strategy, as illustrated in Figure 1.

The input to the recognition method is a single monocular image from the robot camera. The classification method is non-parametric; grasp classification and hand orientation regression is formulated as a problem of finding the hand poses most similar to those in the database. For the database, a large set of examples, from many different views, is provided for each grasp. The details are described in [6].

The human-to-robot grasp mapping scheme depends on the type of robot hand used, two in our current system. The Barrett hand is a three fingered, four degrees of freedom hand with kinematics substantially different to that of a human hand. Our Barrett hand is equipped with tactile sensors on the interior part of the three fingers and on the palm. Karlsruhe hand is a five fingered, eight degrees of freedom hand with kinematics similar to a human hand. The preshapes used for the hands are shown in Figure 2. There are three different grasps for the Barrett hand: the Barrett Wrap, used for grasps with a preshape with large aperture, like Large and Small Diameter grasps; the Barrett Two-finger Thumb, for small aperture preshapes like the Pinch grasp, and the Abducted Thumb (executed as a pinch grasp); and the Barrett Precision Disc, for circular objects. There are four preshapes for the Karlsruhe hand. The Karlsruhe Power preshape is applied for grasps with four parallel fingers and thumb opposed to them, like Large Diameter grasp and Pinch grasp (and Abducted since this grasp cannot be imitated properly). The Karlsruhe Wrap is applied for the Small Diameter where the thumb is not opposed to the rest of the fingers. There are finally two preshapes for round objects, Karlsruhe Sphere (for Power Sphere) and Karlsruhe Disc (for Precision Disc); the differences are in the pose of the thumb (slightly more opposed in the Disc) and in how straight are the rest of the fingers (more bent in Power Sphere).

The hand orientation estimate relative to the camera, along with the hand position estimate and the estimated position and orientation of the grasped object relative to the camera, are used to derive the estimated position and orientation of the human hand relative to the object, as depicted in Figure 1. The estimation of object position and orientation is assumed known.

Different from our previous work, [29], the robot does not explore a range of approach vectors, but instead directly imitates it from the human, as encoded in the hand position and orientation relative to the object.

Based on the estimated type of grasp, the system first differentiates between volar and non-volar grasp ([12]), i.e., whether there should be a contact between the palm and object or not. The original volar grasps are the Large Diameter, Small Diameter, Abducted Thumb and Power Sphere grasps, see Figure 2. However, the hand kinematics make it impossible to use the palm in the Abducted Thumb and Power Sphere grasps. In a human Abducted Thumb grasp, the palm adapts its shape to the object and the abduction/adduction degrees of freedom of the fingers are used; the robotic hands studied here lack those degrees of freedom, so the Abducted Thumb is mapped to the Pinch Grasp. In the case of the Power Sphere, the robotics hand cannot apply a volar grasp due to the differences in size of the fingers. The contact between the palm and the object makes possible to use tactile sensors on the robotic hand to perform corrective movements during the final part of the grasp execution. This makes the grasping less sensitive to object pose estimation errors, as we will show in Section IV.

The volar grasping is performed in the following order: First, the robot adopts the hand orientation and preshape corresponding to the estimated human grasp. The robot hand then approaches the centroid of the object until a contact is detected. After that, it closes the fingers. Two different ways

of approaching the object are used, based on the orientation of the human hand; if the palm plane is approximately parallel to the table plane the object is approached from the top, otherwise it is approached from the side. If the first contact did not occurred in the palm, the hand retracts and the trajectory is replanned. The new goal position for the hand is a weighted average between the detected contact (computed through contact information and kinematics of the robot and the hand) and the original goal position: $p_{new} = \alpha * c + (1 - \alpha) * p_{old}$, where c is the contact position and p the estimation of the object position. The weight α is decreased as $\alpha_{new} = \alpha_{old}^2$ to stabilize the process of goal correction, in case the goal is not reached in the first correction.

The non-volar grasps, which have no contact between the palm and the object, are originally the Pinch grasp and the Precision Disc grasps. Since there is no expected contact between the palm and the object, in our system the grasp is performed open-loop. The robot adopts the hand orientation and preshape based on the human example and the robot hand approaches the object until it is at a predefined distance over the object. After that, it closes the fingers.

IV. Experimental Results

We present grasp mapping both in simulation using GraspIt! and on a real robot equipped with the Barrett hand.

A. Simulated grasping with GraspIt!

GraspIt! is used because it provides the access to different robotic hands and easy control of the object pose so that we can study the effect of pose estimation noise to each of the grasp types.

Evaluating the performance of a grasp imitation system is not trivial. It cannot be based on grasp stability, since the human grasps are not always optimal in terms of stability. A comparison of joint angles between a robot and a human hand is not possible because of the differences in the kinematics and number of degrees of freedom. The contact points on the object may however provide more information about how similar the grasps are, but the differences in the kinematics makes again a direct comparison difficult.

We therefore compare the grasps using the concept of virtual fingers, [13], implemented based on the methodology stated in [12]. The position and orientation of contacts is automatically extracted from the robotic simulator for the robot grasps, and it was tagged manually from images for the human grasp. As cited in Section II, a virtual finger is a group of real fingers, including the palm, that act in unity. In theory, the average position and orientation of the virtual finger contacts in the imitated grasp should be as similar as possible to the ones in the human grasp. However, as it will be discussed later in this section, there are cases where this does not apply.

In the first experiment, object pose is assumed known. Figure 3 represents the grasps and the contact comparison between the robotic hands (black) and the human hand (blue). The big arrows show the position of the virtual fingers for each. It can be seen from the figure that the pose and number of the virtual fingers does not always correspond between the human and the robot hand. For example, the Barrett hand has three virtual fingers for the Small Diameter grasp, while Human and Karlsruhe hands have two, Figures 3a,g,m,s. The reason for this mismatch is that Barrett fingers are longer than human and those of the Karlsruhe hand, so the object is touched by the last phalanx on the edges instead of the face. Another significant difference in the number of virtual fingers appears in the Power Sphere grasp, Figures 3c,k,q,w. The human grasp has just one virtual finger, while the robotic hands have two. For the human, placing the thumb opposed to the rest of the fingers is uncomfortable. The large contact surface, hence large friction, between the hand and the ball allows to place the fingers in a relatively unstable way. However, the contact surface between the robotic hands and the object is much smaller, so the thumb should be placed in opposition to the rest of the fingers. In contrast, in the Precision Disc grasp (Figures 3f,r,l,x) the human needs to place the thumb in opposition to the rest of the fingertips due to the lower friction between the hand and the object. It is also interesting to reason about the results for the Large Diameter grasp, Figures 3b,h,n,t. There is a big difference between the average virtual finger position, but the actual contacts look similar. The reason is that the human fingers have contacts in both the proximal and distal phalanges, while the robot achieves a contact just in the distal ”phalanges”. Finally, it should be pointed out that despite of not imitating properly the Abducted Thumb grasp, the position of the virtual fingers is quite accurate. There is a small deviation in orientation in one of them due to the two contacts from the human palm and index fingertip in the top of the object, which does not exist in the robot grasps.

A quantitative evaluation of the experiments is shown in Figure 4: it represents the average error in virtual fingers position and orientation between the robotic hands (where Barrett error is represented in black and Karlsruhe in white) and the human hand. However, it should be noted that this measure is a lower bound of the error: in cases where the number of virtual fingers is different, this represents the average distance between the best matching virtual fingers. Sometimes this mismatch is known and natural (like the different number of virtual fingers in the Power Sphere grasp), but sometimes this means that one finger failed to touch the object. This happens mainly in the experiments with position error with the Karlsruhe hand, and is explained later on in the section.

For the case of known pose, Figure 4a for orientation, Figure 4e for position, we can conclude that Karlsruhe hand performance in terms of virtual fingers orientation is better. The performance in terms of position of the virtual fingers is similar; the biggest errors appear in the Large Diameter grasp, for the reasons stated before.

We have further evaluated the robustness of the imitation to object position errors. For this purpose, the object pose was varied 5cm in 6 different directions (multiples of $\frac{\pi}{3}$

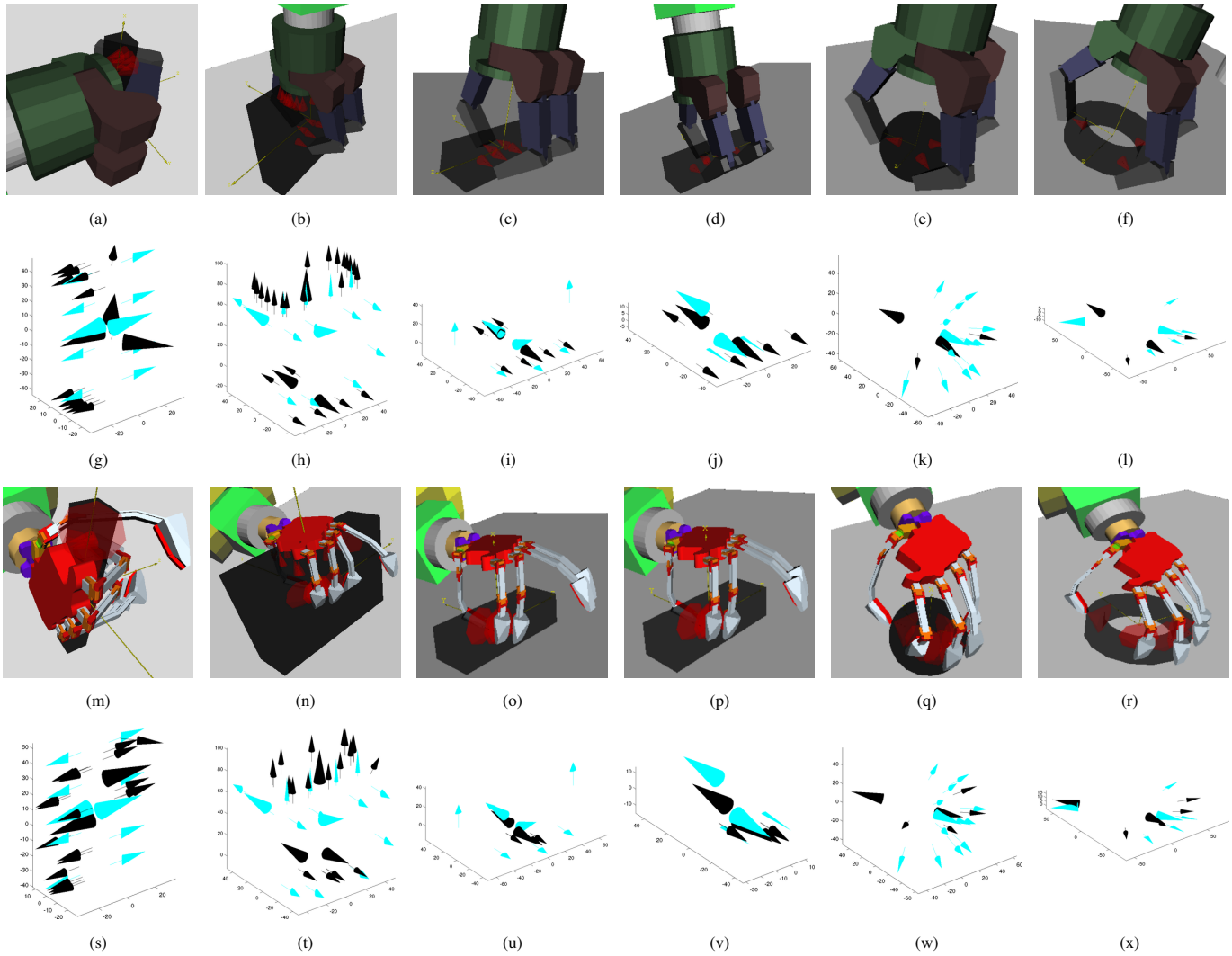
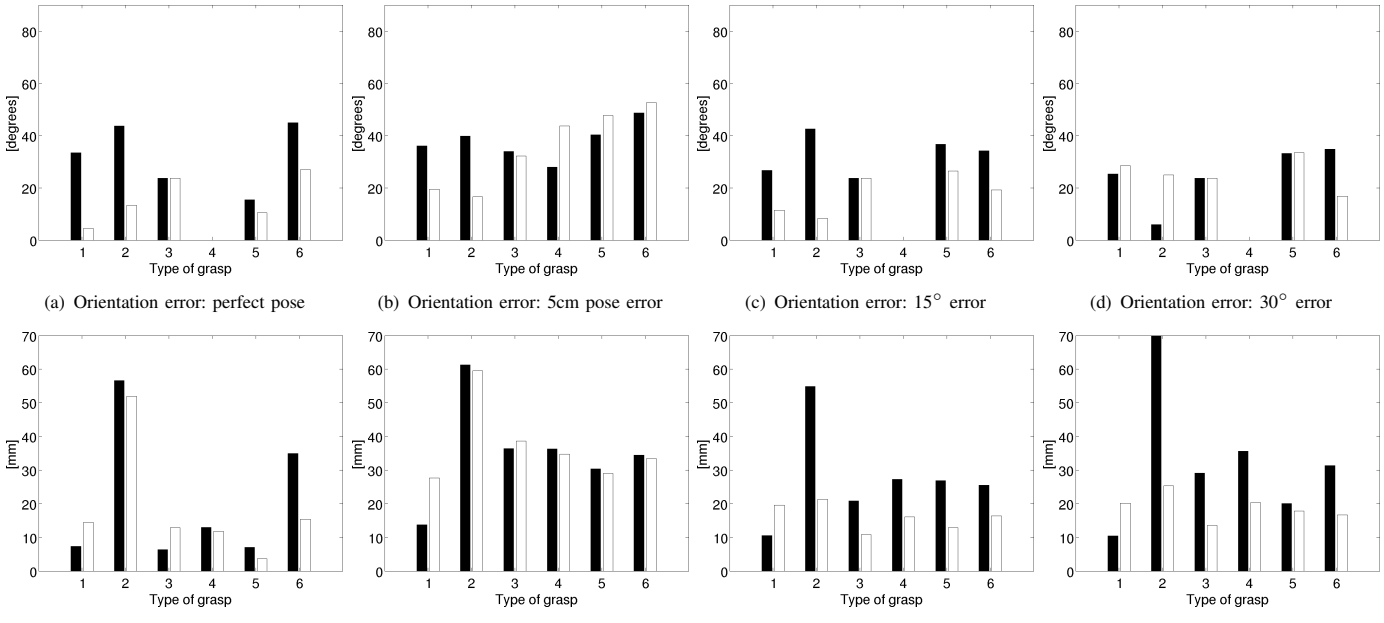


Fig. 3. First and third row shows the grasp execution in absence of errors for Barrett and Karlsruhe Hand. Second and forth row show a comparison between the contacts for Barrett(black)-Human(blue) hands and Karlsruhe(black)-Human(blue) hands. The big arrows show the average pose of the virtual fingers

rad). The difficulty of the problem should be considered first: the size of the objects along the major axis is around 10cm, so the error is significant. Another factor to take into account is the lack of any visual feedback. This experiment shows principally the importance of the feedback (tactile feedback in our case) in the presence of errors. The grasps where tactile feedback was used are the Large and Small Diameter grasps since only for these we expect a first contact to be on the palm: this means that a first contact detected on any other finger is considered to be an error in the object pose that should be corrected. It can be seen that the error increases much more in the grasps without corrective movements (grasps 3,4,5 and 6) than in the ones with corrective movements (grasps 1 and 2). Another thing that can be seen in Figures 4b,f is that the Karlsruhe hand is more sensitive to the errors than the Barrett hand. The error for the Karlsruhe hand in not corrected grasps is higher than the one showed, because the thumb commonly failed to touch the object, and therefore the thumb virtual finger was not compared. There are two reason for this performance: first, the shortest length of the fingers; second, the palm configuration in the Karlsruhe hand. The shortest length of Karlsruhe fingers affect the non-volar grasps, as we can see in Figure 5a,b. The configuration of the palm that has a small distance between the base of the thumb and the base of the rest of the fingers. This affects the volar grasps, that usually collide with the finger bases before touching the palm. However, this is mostly solved by the corrective movements.

Finally, the last two columns of Figure 4 represent the errors for an experiment with rotation error in the object of 15°, Figure 4c,g and 30°, Figure 4c,g. The most important lesson to learn from this group of experiments is that the tolerance to object orientation errors is higher.



(a) Orientation error: perfect pose (b) Orientation error: 5cm pose error (c) Orientation error: 15° error (d) Orientation error: 30° error

(e) Position error: perfect pose (f) Position error: 5cm pose error (g) Position error: 15° error (h) Position error: 30° error

Fig. 4. Error in position (mm) and orientation (degrees) for each of the six grasps tested (Small Diameter, Large Diameter, Abducted Thumb, Pinch, Power Sphere, Precision Disc). Each column represents experiments with no error, error of 5cm in position and error of 15° and 30° in orientation

(a) (b)

Fig. 5. Example of performance of a grasp without corrective movements.

B. Real grasping with a KUKA arm

For the real experiments, the scenario is composed of a robotic stereo head looking at a table where the human performs various grasps to be imitated by the robotic arm and hand (KUKA industrial arm and Barrett hand). The image of the human grasp is captured and passed to the grasp recognition [6], which returns the type of grasp and the position and orientation of the hand. The object pose is given manually. With all this data, the grasp policy is selected and executed.

The scenario, illumination and subject is different to the experiments presented in [6], but we get similar results in the classification. Large diameter, small diameter and abducted thumb are correctly classified most of the time, while pinch grasp, power sphere and precision disc grasp are sometimes confused with the power grasp. In terms of orientation, the typical error is around 15 degrees, which is acceptable in the execution of the grasp, as discussed above.

The object position is given manually, with an error of ± 3 cm. The position error did not affect the grasp execution, except when performing Precision Disc grasp with a ball, which rolled when the hand was not centered over the ball.

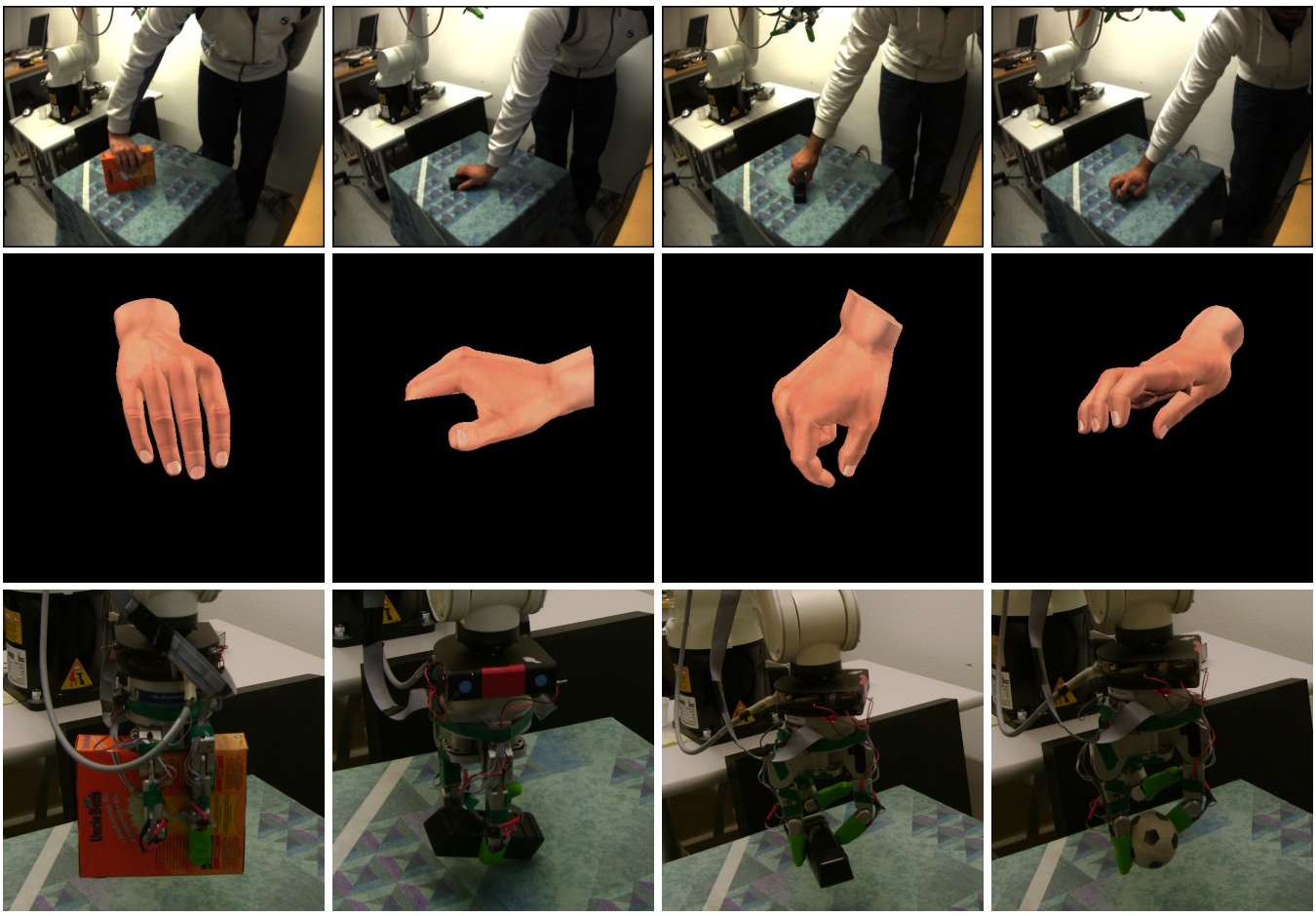
Figure 6 shows the robot being shown four different grasps (Large Diameter, Abducted Thumb, Pinch and Precision Disc, respectively), mapping them and performing the corresponding grasp (Barrett Wrap, Barrett Two-finger Thumb, Barrett Two-finger Thumb and Barrett Precision Disc, respectively).

V. Conclusions

In this paper, a method for grasp mapping based on single view human grasp classification was presented. The grasp classification retrieves the grasp class together with the orientation of the hand. This information was used to select and parameterize a policy for performing the grasp on a robot in a simulated and a real environment. The experiments indicated a need for sensor feedback during the execution of the grasp on the robot.

The main contribution of the paper, apart from the markerless classification strategy, is the use of virtual fingers approach for the evaluation of the mapping quality. This strategy also gives a better insight into the relation and mapping between two very different kinematic chains such as the human and the Barrett hand.

Our current work considers evaluation of the approach on other robotic hands and the use of additional non-parametric regression methods for estimating grasp type and hand pose for a larger set of classes. This will allow us to concentrate on more detailed modeling of grasping strategies related to functional aspects of objects.



(a) 1 → Barrett Wrap. (b) 4 → Barrett Two-finger Thumb. (c) 9 → Barrett Two-finger Thumb. (d) 12 → Barrett Precision Disc.

Fig. 6. Execution of grasps in a real robot environment. First row shows images grabbed by the robotic head at the grasping moment, second row shows the nearest neighbors to the first row pictures in the database, and the third row shows the robot execution of the same grasp. a) Large Diameter grasp,1, mapped to Barrett Wrap. b) Abducted Thumb grasp, 4, mapped to Barrett Two-finger Thumb. c) Pinch grasp, 9, mapped to Barrett Two-finger Thumb. d) Precision Disc grasp, 12, mapped to Barrett Precision Disc.

VI. Acknowledgments

This research has been supported by the EU through the project PACO-PLUS, FP6-2004-IST-4-27657, and by the Swedish Foundation for Strategic Research.

References

[1] V. Kruger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: A review on action recognition and mapping," *Advanced Robotics*, vol. 21, no. 13, pp. 1473–1501, 2007.

[2] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 370–384, 2006.

[3] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, "Enabling real-time full-body imitation: a natural way of m-ansferring human movement to humanoids," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 2368–2374.

[4] P. Azad, A. Ude, T. Asfour, and R. Dillmann, "Stereo-based markerless human motion capture for humanoid robot systems," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 3951–3956.

[5] M. Cutkosky, "On grasp choice, grasp models and the design of hands for manufacturing tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 269–279, 1989.

[6] H. Kjellström, J. Romero, and D. Kragić, "Visual recognition of grasps for human-to-robot mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[7] J. Rehg and T. Kanade, "Visual tracking of high dof articulated structures: An application to human hand tracking," in *European Conference on Computer Vision*, vol. 2, 1994, pp. 35–46.

[8] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, "A hand-pose estimation for vision-based human interfaces," in *IEEE Transactions on Industrial Electronics*, vol. 50(4), 2003, pp. 676–684.

[9] V. Athitsos and S. Sclaroff, "Estimating 3D hand pose from a cluttered image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 432–439.

[10] C. Schwarz and N. Lobo, "Segment-based hand pose estimation," in *Canadian Conf. on Computer and Robot Vision*, 2005, pp. 42–49.

[11] H. Murase and S. Nayar, "Visual learning and recognition of 3-D objects from appearance," *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.

[12] S. B. Kang and K. Ikeuchi, "Toward automatic robot instruction from perception: Mapping human grasps to manipulator grasps," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 1, pp. 81–95, 1997.

[13] M. A. Arbib, T. Iberall, and D. M. Lyons, "Coordinated control programs for movements of the hand," in *Hand function and the neocortex. Experimental Brain Research Supplemental 10*, A. W. Goodwin and I. Darian-Smith, Eds., 1985.

[14] M. Jeannerod, "Intersegmental coordination during reaching at natural visual objects," *Attention & Performance*, vol. IX, 1981.

[15] A. M. Wing, A. Turton, and C. Fraser, "Grasp size and accuracy of approach in reaching," *Journal of motor behavior*, vol. 18, 1986.

[16] A. Morales, P. Azad, T. Asfour, D. Kraft, S. Knoop, R. Dillman,

A. Kargov, C. Pylatiuk, and S. Schulz, "An anthropomorphic grasping approach for an assistant humanoid robot," in *International Symposium of Robotics*, 2006.

[17] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1824–1829.

[18] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Neural Information Processing Systems*, 2006.

[19] K. Hübner and D. Kragić, "Selection of robot pre-grasps using box-based shape approximation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[20] C. Borst, M. Fischer, and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3D objects," in *IEEE International Conference on Robotics and Automation*, 1999, pp. 1890–1896.

[21] Y. Li and N. Pollard, "A shape matching algorithm for synthesizing humanlike enveloping grasps," in *IEEE International Conference on Humanoid Robots*, 2005.

[22] S. Ekvall, "Robot task learning from human demonstration," Ph.D. dissertation, KTH, Stockholm, Sweden, 2007.

[23] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele, "Functional object class detection based on learned affordance cues," in *Computer Vision Systems*, 2008.

[24] A. M. Borghi, "Object concepts and action," in *The Grounding of Cognition: The role of perception and action in memory, language, and thinking*, D. Pecher and R.Zwaan, Eds. Cambridge University Press, 2005, part 2, pp. 8–34.

[25] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," in *International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, 1999, pp. 103–115.

[26] K. Ogawara, J. Takamatsu, K. Hashimoto, and K. Ikeuchi, "Grasp recognition using a 3D articulated model and infrared images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2003, pp. 1590–1595.

[27] L. Y. Chang, N. S. Pollard, T. M. Mitchell, and E. P. Xing, "Feature selection for grasp recognition from optical markers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[28] J. Triesch, J. Wieghardt, E. Mael, and C. Malsburg, "Towards imitation learning of grasping movements by an autonomous robot," in *International Gesture Workshop*, 1999, pp. 73–84.

[29] J. Tegin, S. Ekvall, D. Kragić, B. Iliev, and J. Wikander, "Demonstration based learning and control for automatic grasping," in *International Conference on Advanced Robotics*, 2007.