Predicting Human Intention in Visual Observations of Hand/Object Interactions

Dan Song, Nikolaos Kyriazis, Iason Oikonomidis, Chavdar Papazov,
Antonis Argyros, Darius Burschka and Danica Kragic

***Abstract*— The main contribution of this paper is a probabilistic method for predicting human manipulation intention from image sequences of human-object interaction. Predicting intention amounts to inferring the imminent manipulation task when human hand is observed to have stably grasped the object. Inference is performed by means of a probabilistic graphical model that encodes object grasping tasks over the 3D state of the observed scene. The 3D state is extracted from RGB-D image sequences by a novel vision-based, markerless hand-object 3D tracking framework. To deal with the high-dimensional state-space and mixed data types (discrete and continuous) involved in grasping tasks, we introduce a generative vector quantization method using mixture models and self-organizing maps. This yields a compact model for encoding of grasping actions, able of handling uncertain and partial sensory data. Experimentation showed that the model trained on simulated data can provide a potent basis for accurate goal-inference with partial and noisy observations of actual real-world demonstrations. We also show a grasp selection process, guided by the inferred human intention, to illustrate the use of the system for goal-directed grasp imitation.**

I. Introduction

Humans possess a profound capability of learning though imitation. Imitation learning has been addressed frequently in robotics for enabling bootstrapping in object grasping and interaction with the environment in general, [28], [8], [20], [27], [2]. An important challenge in imitation learning is the "correspondence problem" [21] due to the differences in embodiments between humans and robots, being especially problematic in grasping applications [32], [30], see an example in Fig. 1. One solution to the correspondence problem is "goal-directed imitation" [34] where the learner infers the intent of the activity from the teacher and then achieves the same goal using own sensorimotor acts (see a flowchart of goal-directed imitation process in the context of robot grasping in Fig. 6). Recognizing the intention of object grasping actions in humans is then an integral part of enabling robots to act and interact with the environment. The intention in the scope of object grasping and manipulation is what kind of tasks a human / a robot intend to do after sucessfully grasping the object. An example of such high-level tasks is to hand-over an apple as shown in Fig. 1. Most of the work that addressed this problem is either evaluated in simulated environments where perfect knowledge of objects and grasps is available [32], [30], or in situations where the human teacher is equipped with magnetic sensing or datagloves [17], [5].

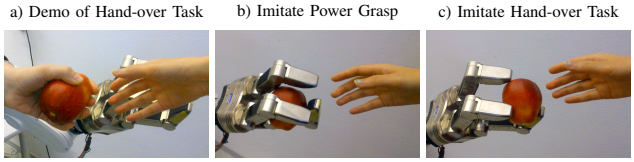

Fig. 1. a) Human demonstrates a grasp with an intention to *hand-over an apple*. b) Robot imitates the power grasp configuration used by the human, and fails to hand-over because there is not enough free space for regrasp. c) Robot estimates that human intends to hand-over the apple. It also learns the task requires leaving enough free space on the object. It applies a precision grasp to achieve the task.

In this work, we show how to approach the problem of learning object grasping and manipualtion tasks by considering natural human demonstration - with no markers and no datagloves. A similar idea was also exploited in [11], [12], [33]. In [12] the recognition module was based on a discriminative model that could not be used to generate grasping actions in the imitation phase. [11] addressed grasp imitation on a robot platform but the approach was based on manual mapping between human and robot hands, thus difficult to generalize across different embodiments. The work in [33] focused on extracting object affordances [7]. However, detailed grasp parameters such as hand position, orientation and finger configuration were not considered. We argue that these are actually the most important parameters that encode the information relevant for the task. Work in [32], [30] adopted a generative modeling approach similar to us. However they only tested their model in simulation environments. We present our state of the art, markerless, vision-based 3D hand and object tracking system that is capable of extracting detailed parameters of human hand motion in interaction with objects.

To estimate human intention, we build a model that encodes the grasping tasks (i.e., intentions) using probabilistic graphical models as in [32], [30]. We show how to model

D. Song and D. Kragic are with the KTH – Royal Institute of Technology, Stockholm, Sweden, as members of the Computer Vision & Active Perception Lab., Centre for Autonomous Systems, www: http://www.csc.kth.se/cvap, e-mail addresses: {dsong,chek,khubner,danik}@csc.kth.se.

N. Kyriazis, I. Oikonomidis and A. Argyros are with the Foundation for Research and Technology, Crete, Greece, as members of Institute of Computer Science, www: http://www.ics.forth.gr/cvrl/, e-mail addresses: {kyriazis, oikonom,argyros}@ics.forth.gr and with the University of Crete, Crete, Greece, as members of the Computer Science Departnment, www: http://www.csd.uoc.gr/, e-mail addresses: {kyriazis, oikonom,argyros}@csd.uoc.gr.

Chavdar Papazov and Darius Burschka are with Technische Universität München, Munich, Germany, as members of Machine Vision and Perception Group, www: http://www6.in.tum.de/burschka/, e-mail addresses: {papazov, burschka}@cs.tum.edu.

the object grasping domain using a combination of discrete Bayesian networks and vector quantization models. The generative modeling approach allows goal inference under partial and noisy observations in real world applications. Combined with the markerless vision-based hand/object tracking module, we show that the generative modeling approach results in accurate goal inference even under partial and noisy observations in real world applications. Finally we illustrate the applicability of the learned model in a goal-directed imitation scenario.

II. Models

A. Modeling Grasping Tasks using Graphical Models

In this section we shortly describe the use of BN for grasp planning and learning of goal-directed grasping policies. Goal-based grasp planning involves decisions over (a) object selection and (b) grasp realization so that a given task, out of several, can be afforded. Let also T be a selection over common house-hold task set $\mathcal{T}$ that an agent might intend to do after grasping an object. Let O be the set of variables describing object category, size, and shape-related features. Let A be the set of variables that parameterize a grasp action such as hand position, orientation and articulation. Planning a grasp is then to decide O and A given T. In addition to O, A, we also introduce a set of "constraint" variables, denoted as C that further describe the configuration of the object and hand in the final object-grasp complex. An example C variable is free volume that quantitize how much free space on the object is left uncovered by the hand in final grasp configuration.

We assume that object related variables O are intrinsic properties (they do not change as the object moves) and action variables A regard a static hand pose, at the moment of grasping, expressed in the local coordinate frame of objects. A variables are agent-specific because humans and robots differ in the kinematics of their end-effectors. Each object-grasp complex, as shown in Fig. 1, can be described by a set of object, action and constraint variables O, A, C and the task(s) T that the grasp can afford. While O, A, C variables can be measured or extracted using robot's vision and haptic sensors, the high level task semantics such as *"grasp a water bottle in order to pour water"* or *"grasp the cup to put it in dishwasher"* have to be provided by a human teacher. Therefore while O, A, C is a set of variables that are instantiated by robot's sensor measurements, T is instantiated by human labeling that states if this grasp could afford the given task label.

If a large set of example grasps with task semantics is provided, we can train a BN to model the joint distribution of all variables $p(O, A, C, T)$. We use BN^R to represent BNs for a robot and BN^H for humans. Goal inference is then the process that determines the most likely task(s) t^* through inferring the marginal probability $p(t|\mathbf{o}, \mathbf{a}, (\mathbf{c}), BN^H)$, given a grasp demonstration with observed object features $\mathbf{o}$, grasp action parameterized by $\mathbf{a}$, and with or without the constraints $\mathbf{c}$. Grasp planning is a process that determines the most appropriate object $\mathbf{o}^* = \arg\max_{\mathbf{o}} p(\mathbf{o}|t^*, BN^R)$, and then the most appropriate grasp $\mathbf{a}^* = \arg\max_{\mathbf{a}} p(\mathbf{a}|\mathbf{o}^*, t^*, BN^R)$ for executing the given task.

B. Generative Soft Vector Quantization

To correctly encode a grasping task, both symbolic information (e.g., task and object class) and continuous low-level sensorimotor variables (most O, A, C features) are needed. Learning a BN from both continuous and discrete data simultaneously is an open problem, particularly for the cases of high dimensionality and complex distributions (e.g., hand grasp configuration and hand orientation). Most learning approaches only work with discrete variables and multinomial distributions [15]. An efficient approach is to discretize a mixed model by quantizing the continuous variables [6]. In the common case where quantization incurs "hard" cuts between neighboring discrete states, each continuous data point x is mapped to a single discrete state. This is disadvantageous especially in a domain with limited training data, obtained from noisy observations. This problem has been identified and remedied in [16], [4] with "soft" discretization, i.e. by taking into account the influence of a continuous data point over its neighboring discretization intervals. The resulting Bayesian network can be interpreted as a "fuzzy" network. Such solutions have been shown to outperform their hard-discretization counterparts [16], [4].

We use a generative soft vector quantization approach for our high-dimensional continuous problem, based on self-organizing maps (SOM) and Gaussian mixture models (GMM). Here "fuzziness" in the discrete assignment of continuous data is not handled using a spread function convoluted with the manually-defined discrete intervals as in [4]. Instead, clustering is employed to automatically define the boundaries of discrete intervals, thus can efficiently cope with high-dimensional data. Given the initial clustering results, we then obtain a GMM model to represent the density of each input space. GMM provides a probability representation that can "smooth out" the boundaries in the initial clusters (see Sec. II-B.1 and Sec. II-B.2). GMM model is also used in [16] to obtain a fuzzy BN. Different from [16] who integrated GMM as a node in the Bayesian network, our GMM is decoupled from the network. Each continuous data is discretized prior to training of the BN. This increases training efficiency compared to [16].

1) Clustering using Self-Organizing Map: For a high-dimensional continuous variable Z, we use a SOM-based clustering approach as in [35] to form K clusters on a dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, where N is the number of samples. The principle is to first use SOM to project $\mathbf{X}$ to a set of prototypes (map units) that are then combined to form final clusters.

SOM [13] is often used in quantization and visualization of high-dimensional data and it consists of a regular, usually 2-dimensional grid of map units. Each unit j is represented by a prototype vector $\mathbf{w}_j$ that has the same dimensionality as the original data. The units are connected by neighborhood relations discovered from original training data.

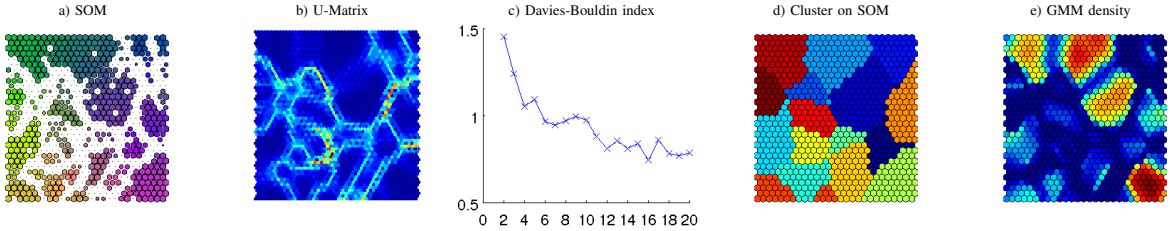

Fig. 2. SOM-based Soft Discretization illustrated on the 4D data of the Quaternion Hand Orientation: a) The SOM with the size of the unit determined by number of original training data that is mapped to that unit; b) U-matrix where the the light color indicates large distance between adjacent map units; c) Davies-Bouldin index the minimization of which determines the optimal number of the clusters K; d) the clustering results on the SOM; e) the density map from the trained GMM.

The SOM can thus be interpreted as a topology preserving mapping from input space $\mathbf{X}$ onto the 2D map units $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M]$ resembling the density of the original data. Given this property, the problem of clustering on the original N data points can be reformuated into clustering on their mapped M prototype vectors on the SOM. Since $M << N$, clustering is more efficient in computation.

Fig. 2 illustrates this process. First, a large set of prototypes – much larger than the expected number of clusters K– is formed using the SOM (Fig. 2 a). The prototypes are combined in the next level to form the actual clusters (see Fig. 2 d). The U-Matrix (Fig. 2 b), which represents the distance between neuron weights, can be used to determine the clusters and their boundaries. Given a predefined number of clusters K, clustering is performed on $\mathbf{W}$ using the K-means algorithm. To determine K, we use the Davies-Bouldin validity index, defined as $\mathcal{I}_{db} \triangleq \frac{1}{2} \frac{1}{K} \sum_{k=1}^{K} \max_{l \neq k} \{\frac{s_k + s_l}{d_{kl}}\}$, where s_k is intra-cluster distance and d_{kl} the inter-cluster distance. K is the minimizer of $\mathcal{I}_{db}$ (see Fig. 2 c). Each data point of the original data set $\mathbf{X}$ is assigned to the cluster of its prototype $\mathbf{w}_j$.

2) Gaussian Mixture Model: The result of the first step is a selected number of clusters K and the corresponding cluster assignment on the original data $\mathbf{X}$. This result allows for the learning of a Gaussian mixture model that encodes the density distribution in the original data space (see Fig. 2 e),

$$p(\mathbf{x}) \propto \prod_{k=1}^{K} \lambda_k N(\mathbf{x}|\mathbf{u}_k, \Sigma_k^{-1}). \tag{1}$$

where $\mathbf{u}_k$ and Σ_k are the mean and covariance of each Gaussian component, and λ_k are the mixing weights. The parameters of the mixture model are learned using a standard EM approach.

Given the GMM model, a continuous data point $\mathbf{x}$ is converted to a soft discrete evidence $\mathbf{y} = [y_1, y_2, ..., y_K]^T$, where $y_k = p(\mathbf{x}|k)$ is the probability of the data being generated by the k^{th} mixture component in GMM . These probabilities quantify the level of influence of a continuous data point over its neighboring discretization intervals. In parameter learning of the BN, we can then update the conditional probability table for each variable using the soft discrete evidences (details of the algorithm refer to [4]).

C. Inference

A trained BN defines a factorization of the joint distribution $p(T, O, A)$, that adheres to the discovered conditional dependencies. The availability of such a BN enables the computation of the marginal probability density of a group of variables by conditioning on the rest. This is commonly performed by applying the junction tree algorithm [9]. The result of the latter is a set of probabilities, for all the discrete states of a variable Z, conditioned on the observed evidence e: $p_k = p(Z = k|e)$. The expected continuous value $E(Z)$ of the variable Z is required.

Estimating a continuous representation from soft evidence is an "inverse problem" in soft discretization as it recovers the continuous value from discrete states. This is also known as de-fuzzification in fuzzy set theory, which does not have a simple solution [14]. A practical approach for interpreting the output of a discrete Bayesian network is to use the *most likely point* of the inferred variable as the output. We can define a distribution over the continuous space $\mathbf{x}$ associated with the multi-nominal distribution p_k,

$$p(\mathbf{x}|e) \propto \sum_{k=1}^{K} p_k N(\mathbf{x}|\mathbf{u}_k, \Sigma_k^{-1}) \tag{2}$$

where $\mathbf{u}_k$ and Σ_k^{-1} are the mean and covariance of the components of the GMM model (Eq. 1). We can then sample from the above distribution in order to find the most likely locations over the continuous space. This approach will be henceforth referred to as ML. In ML, however, there is a danger of information loss, especially when p_k does not have a strong preferred single state, but it spreads out to multiple states. An alternative method would be to estimate the expected value using a weighted sum of the component centers that takes probabilities of all the states into account. The expected value of variable Z is then defined as,

$$E(Z) = \sum_{k=1}^{K} p(Z = k|e)\ \mathbf{u}_k = \sum_{k=1}^{K} p_k\ \mathbf{u}_k \tag{3}$$

This approach is expected to be better if the different modes (state centers) preserves the topological map of the original data space

III. Vision-based Object and Hand Pose Tracking

In order for inferring the intention given natural observations, the latter need to be translated into the domain of the

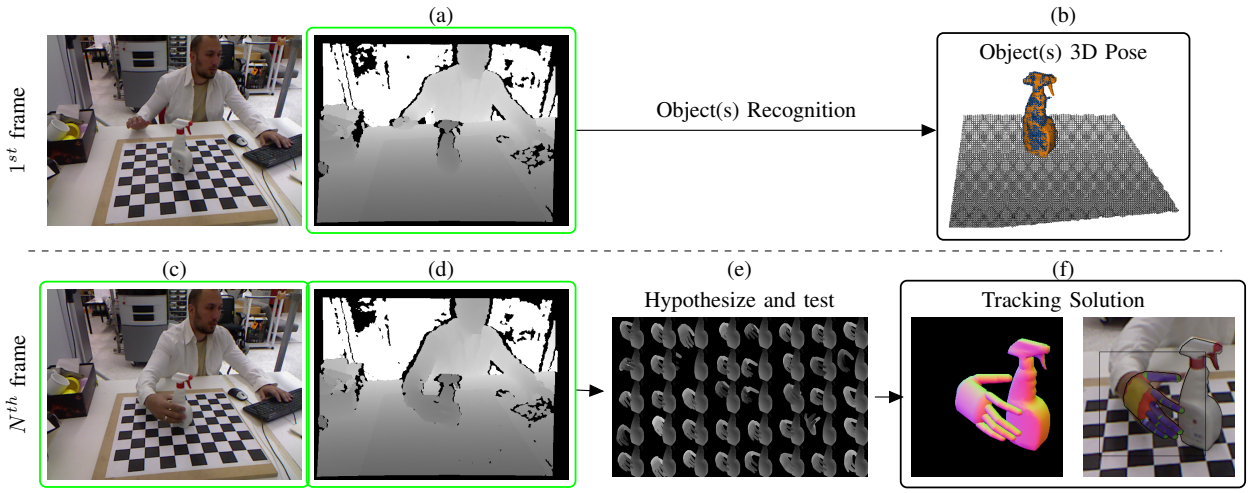

Fig. 3. Tracking: The first frame of a sequence is used to recognize objects of a database that appear in a scene, according to Sec. III-A. The rest of the frames of the hand-object(s) ensemble are tracked in a hypothesize-and-test fashion (see Sec. III-B). The channels of information of each frame that are used in each phase are outlined with green borders.

proposed probabilistic model. In our work, the observations are acquired using a RGBD sensor [18]. Translation is achieved by means of a novel hand-object 3D tracking system that combines state-of-the-art methods for object recognition [25], [26] and hand-object 3D tracking [23], [22].

At any given time instant, the state of a human's hand and the object(s) he/she interacts with are represented by a vector that encodes the involved degrees of freedom (DoFs). For a hand 26 parameters are required to represent its pose and articulation [22]. For an object 6 parameters are required to represent its pose (3D position and 3D orientation). Tracking a hand interacting with N object means to extract $26 + 6N$ parameters for each time instant that best explain the actual observations. We formulate this scene tracking as an optimization problem and follow the hypothesize-and-test scheme of [23], [22] while combining their merits. Since tracking is formulated as an evolution of hypotheses over time, an initial state needs to be provided. We delegate this task to the robust and powerful method of [25], [26] and thus fill the gap of [23], [22] with respect to the initialization problem. The tracking process is illustrated in Fig. 3. The reference coordinate system is conveniently defined to reside on the demonstration table rather than inside the RGBD sensor (default). This is achieved by means of straightforward calibration, using a chessboard pattern of known dimensions and structure.

A. Object Recognition and Pose Estimation

Given a set of 3D object models and a scene observation (Fig. 3 a), we need to estimate the position and orientation of the objects present in the scene (Fig. 3 b) in order to initialize tracking. The employed object recognition algorithm [25], [26] consists of an offline pre-processing phase and an online recognition step. In the offline phase, pairs of points with corresponding normals are sampled from the model and a geometric descriptor signature is computed for each pair and stored in a hash-table. The complete representation for all models is computed by processing each model in this way using the *same* hash table. Thus, new models can easily be added without recomputing the whole hash table. In the online recognition phase, a point pair is sampled from the input scene, the two normals are estimated and the same geometric descriptor is computed. Next, the descriptor signature is used as a key to the hash-table to retrieve all model point pairs similar to the one sampled from the scene. Note that two pairs of points with normals are sufficient to compute a unique rigid transform which aligns them. This results in a number of hypotheses each one consisting of a model and a rigid transform which maps it to the scene. A hypothesis is accepted as a solution if it passes several verification stages. This process of sampling, hypothesis generation and verification is repeated a number of times such that all objects in the scene are recognized with a certain user-defined probability.

B. Tracking

By combining the efficient hand-tracking from RGB-D input of [22] with contextual information over objects under manipulation as in [23], we derive a system that is able to simultaneously track a hand and several objects from RGB-D input. Given an initial recognition of N objects in the scene, tracking of the $26 + 6N$ parameters that best explain the input sequences is performed. As in [23], [22], tracking amounts to a series of optimization steps that correspond to individual time instants along the input sequences.

For a given instant in time, every hypothesis over the $26 + 6N$ parameters yields a 3D configuration of a 3D hand and objects. Upon its 3D rendering, with respect to camera-world calibration information, the hand-object(s) ensemble provides feature maps (depth and silhouettes) that are directly comparable to actual observations (see depth map in Fig. 3 d). The Particle Swam Optimization (PSO) search heuristic is responsible for generating and improving hypotheses over the true state that is captured in the observations. A hypotheses is improved if it is altered so as to score better in an objective function that quantifies discrepancies between 3D renders

TABLE I

FEATURE DESCRIPTION.

	Name	Dim	Description
T	$task$	-	Task Identifier
O_1	$obcl$	-	Object Class
O_2	$size$	3	Object Dimensions
O_3	$conv$	1	Convexity Value $[0, 1]$
A_1	dir	4	Approach Direction (Quaternion)
A_2	$npos$	3	Unit Grasp Position
A_3	$fcon$	20	Final Hand Configuration
C_1	$fvol$	1	Free Volume
C_2	coc	3	Center of Contacts
C_3	$gbvl$	3	Grasped Box Volumn

of it and actual observations. In this paper, instead of only considering a 3D hand model during 3D rendering (as in [22]), we also consider the recognized 3D objects, which additionally modulate the objective values. The availability of additional appearance models over predefined objects allows more direct incorporation of the objects during tracking than in [23]. These models are defined, trained in an offline phase and used online according to [1].

IV. EXPERIMENTS

Consider a setup where a robot observes a human executing a task that involves grasping and manipulation of objects. The goal of the robot is to (1) infer the intention of the teacher given a demonstration example and (2) to reach the intended goal using its own grasping policy. To do this, the robot has to learn from past observations of human demonstration and past experiences of its own grasp execution. These knowledge are encoded in human-specific networks BN^H, and networks specific to its own embodiment BN^R.

The experiments in this section are designed to first evaluate the accuracy in goal inference using BN^H trained with simulated data. We then test the ability of this model to recognize the goal of real human grasps tracked with the proposed vision system in Section III. Finally we illustrate the process of goal-directed imitation on a robot platform using one examplar demonstration. Notice that this illustration is not intended to evaluate the imitation performance on a robot platform, but to place the proposed intention prediction method in a full goal-directed imitation framework in order to demonstrate how to apply it. In the experiments, we consider a set of four tasks: $\mathcal{T} = \{$hand-over, pouring, tool-use, and dish-washing$\}$, which represent common house-hold tasks which we would like a personal robot to help out in a future home.

A. Data Generation

Tab. I show the features used in the experiment. Some variables such as free volumn $fvol$ and grasped box volumn $gbvl$ can be useful to discriminate tasks such as hand-over. However they are not directly observable in real human demonstration. We therefore use simulated data with full observation to train the model, and then test the goal inference performance using both simulated test data and the test data from real human demonstration.

TABLE II

DATA STATISTICS.

	Simulated		Real	
Task	Pos.	Neg.	Pos.	Neg.
Hand-over	996	996	324	636
Pouring	860	860	773	893
Tool-use	986	986	399	336
Dish-washing	457	457	682	893

We use a box-based grasp planner *Box Approximation, Decomposition and Grasping* (BADGr) [10] as a tool to generate the simulated data set that instantiate the $\{O, A, C, T\}$ features as listed in Tab. I. Two hand models – a 20-DoF human hand model and a 7-DoF Schunk Dexterous hand model – are used to generate grasp hypotheses over 48 object models with 6 types: bottle, glass, mug, knife, hammer and screwdriver. To extract those features, we first generate grasp hypotheses using the grasp-planner BADGr [10], and then evaluate them as scenes of object-grasp configurations in a grasp simulator, GraspIt! [19] for task labeling. BADGr includes extraction and labeling modules to provide the set of variables presented in Tab. I. Notice that one object-grasp configuration can be good for multipel tasks among the set $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$. To extract a set of negative examples for a given task t_i, we look through all the grasp instances and pick out the ones that are not valid for t_i. As a result, we may not obtain balanced positive versus negative training set for all the tasks.

The data from real human demonstration is obtained using the vision-based object-hand tracking system described in Section III. Three known (3D models available) objects – a bottle, a mug and a hammer are used for data collection. For each object, a human subject is asked to perform a number of grasps that satisfy one specified task (positive examples) and a number of grasps that are not good for one specified task (negative examples). We finally obtain a set of task-labeled real grasp data that instantiate the $\{O, A, T\}$ features in Tab. I. Notice that even though the vision-based tracking system can provide temporal data during the entire sequence of reaching and grasping action, we only take the data at the last time frame corresponding to the moment of the final stable grasp configurations that are modeled by the static Bayesian networks. Tab. II summarizes the number of the positive and negative examples for each of the four tasks (or intentions, goals) in the simulated and real datasets.

B. Model Training

Given the simulated data $\mathbf{D} = [\mathbf{O}, \mathbf{A}, \mathbf{C}, \mathbf{T}]$, we obtain four task-specific Bayesian networks for the four tasks.

1) Data Discretization: As stated in Section II-B, to learn the structure of the Bayesian network from data, we need to discretize the continuous data. For one-dimensional continuous variables, we first apply the equal-width binning to cluster the continuous data into K intervals, and then train a GMM model based on this initial clustering. The free parameter K is selected from a given range of Ks that minimize the Bayesian information criteror (BIC). For

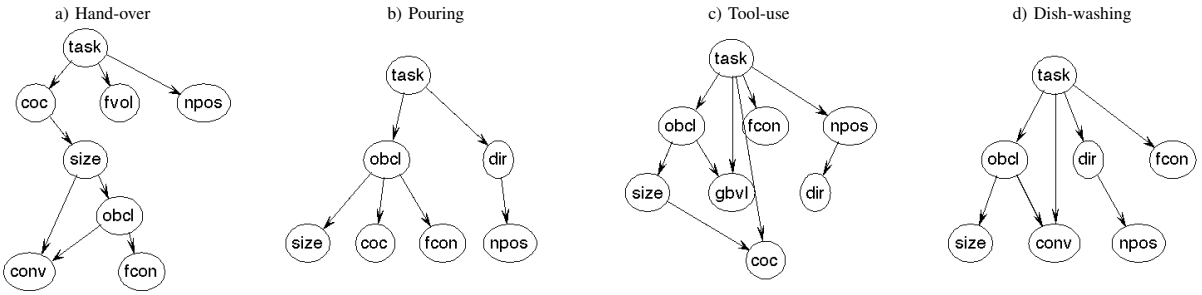

Fig. 4. Task-specific Bayesian networks for human hand. The DAGs shown on a) – d) are learned from the simulated data.

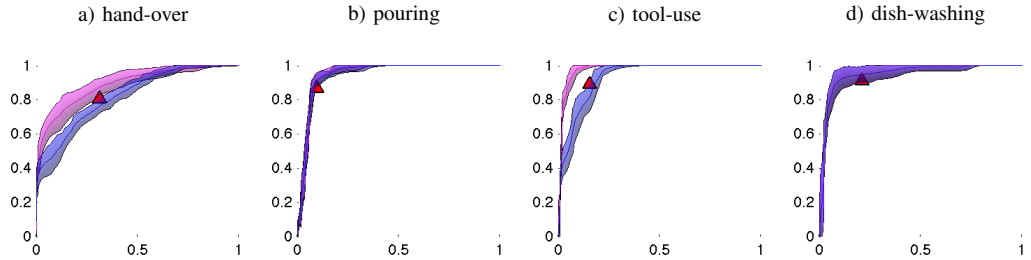

Fig. 5. Goal inference: a) – d) show the average ROC curves (true positive vs. false positive rates) for the four task goals. The shaded region represents 2 standard deviation on true positive rate (Y axis). Two colors represent two different observation conditions: pink – full; blue – partial where *free volumn* and *grasped part shape* are missing. e) summarizes the area under the ROC curves (AUC). Goal inference results on the real human data are shown by the red triangles superimposed on ROC curves. The result is at the optimal classification threshold.

multivariate variables we apply SOM-based discretization. Number of cluster K is determined by minimizing the the Davies-Bouldin (see description in Section II-B).

2) Learning Bayesian Networks: Given the discretized data, we use a greedy search algorithm to find the network structure (the directed acyclic graph, or DAG) in a neighborhood of graphs that maximizes the network score (Bayesian information criterion [29]). The search is local and in the space of DAGs, so the effectiveness of the algorithm relies on the initial DAG. As suggested by [15], we use another simpler algorithm, the maximum weight spanning tree [3], to find an oriented tree structure as the initial DAG. Once the structure is determined, the conditional probability tables are updated sequentially using a standard Bayesian parameter updating scheme.

Fig. 4 shows the learned BNs of the four tasks for the human hand. We observe that the learned network structures successfully reflect the different requirements of the four tasks. For example, for hand-over task, the variable free volumn $fvol$ directly encodes the task requirements, therefore it is directly linked as a child node of the $task$ variable. Since one can hand-over any types of object, therefore the object class $obcl$ is further away from $task$. On the contrary, since other tasks demands specific object types, e.g., a tool-use task needs to select tool objects, $task$ has direct dependency on $obcl$. In additon, we notice that the pose $(npos, dir)$ of the hand relative to the object is important variables to encode task affordance of a grasp. For example, pouring task requires a hand to be placed on the side of a bottle or a mug with specific orientations.

C. Goal Inference

Using the graphical model, the problem of goal inference reduces to finding the marginal $p(t|\mathbf{o}, \mathbf{a}, (\mathbf{c}), BN^H)$, where t is the goal, $\mathbf{o}, \mathbf{a}$ and $\mathbf{c}$ are the observed object features, the grasping action parameters and constraint features when the human hand is closed around the object. This probability can be efficiently computed using inference methods explained in Section II-C. A grasp is considered to be good for one or more tasks $\{t\}^*$ if p is higher than a chosen threshold. If a single goal is to be selected, then it will be the one with the highest probability $t^* = \arg\max_t p(t|\mathbf{o}, \mathbf{a}, (\mathbf{c}), BN^H)$.

To evaluate the performance of goal inference, we show the Receiver Operator Characteristic (ROC) curves, one for each goal, in Fig. 5. The ROC curves are obtained using the simulated data. We use a 30-trial hold-out cross-validation with 70% of the data for training and 30% for testing. Total number of data points for each goal is shown in Tab. II. Each trial induces a ROC curve for one goal when $p(t|\mathbf{o}, \mathbf{a}, BN^H)$ is thresholded at different levels. We then average all ROC curves to obtain the mean ROC curves as shown in Fig. 5. The shaded region represents two standard deviations of the true positive rate. ROC curves are obtained under two observation conditions: 1) full observation is when all O, A, C features are available, and 2) partial observation is when C features are missing which is the situation in real human demonstration.

We observe that the goal inference is quite good for all the four goals even under partial observations (above 0.80 AUC). Hand-over is more difficult than others since it is a less constrained task. To hand-over an object, the object can

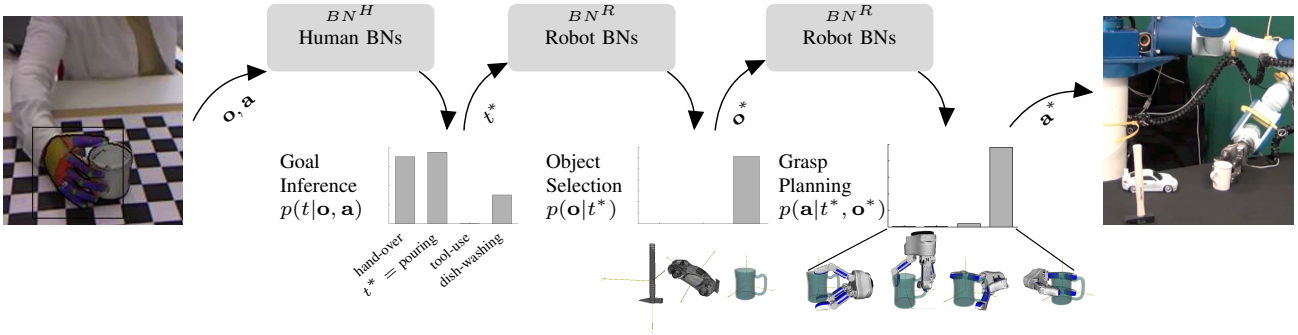

Fig. 6. Flowchart showing the full process of goal-directed grasp imitation. Robot is equiped with the knowledge of human grasp intentions (modeled in BN^H) and its own grasp capability (modeled in BN^R). Given its observation of a human grasp demonstration, it first uses BN^H to infer the intention of the human, and then decides, using BN^R, which object and action that are most suitable to achieve the same intention on its own embodiment. The image of the real robot platform is on the Humanoid torso, TOMBATOSSALS from the Robotic Intelligence Lab at Universitat Jaume I.

be grasped from all directions, but in pouring, the hand can not cover the opening. To achieve the goal of hand-over, a grasp should not cover the entire surface of the object, thus it should induce larger free volume. So when $fvol$ observation is missing, we observe a performance drop in the blue ROC curve. Similarly, a tool-use task requires grasp to be on the handle part of an object such as a hammer, therefore when the part-relevant observation $gbvl$ is missing, we observe worse performance in goal inference.

The most interesting results is when we tested the goal inference with real demonstration using the model trained with the simulated data. Here the observation is partial where information of constraint variables is missing. The red triangles superimposed on the ROC curves on Fig. 5 show the result at the optimal classification thresholds. We observe that goal inference on real human data is as good as the results on the simulated data. This confirms the ability of our model to generalize to real-world data.

D. Goal-directed Imitation

Given the estimated goal t^*, grasp imitation is accomplished by executing the grasping strategy (selecting the object to grasp and how to grasp it) according to the learned grasping policy suitable for the robot's own embodiment BN^R. We train task-specific Bayesian networks for robot hands in the same way as for human hands. Due to space limits, we will not show the learned robot hand BNs. Fig. 6 illustrates the whole process of goal-directed imitation. Note that this is not a real time implementation but a simple flow chart to demonstrate how to use the BN-based grasp reasoning system.

First, the robot observes the human demonstration, and extracts the object and action parameters $\mathbf{o}, \mathbf{a}$ (using its perception system described in Section III). It uses the human-specific networks BN^H to infer the intention of the human, and estimates the most likely goal is t^* =*pouring* with the mug. Second, the robot needs to select the most suitable object for pouring task among the three novel objects on the table: a hammer, a toy car and a mug. This requires the use of the robot's own experiences encoded in BN^R to infer $p(\mathbf{o}|t^*)$. The most likely object is the mug.

Finally, the robot needs to search for the best grasp strategy on the mug that affords pouring. This step can be done in multiple ways. A grasp hypothesis is usually parameterized by three action parameters $A = \{npos, dir, fcon\}$. One way to plan a grasp is to infer the joint distribution of this set of parameters given selected object and the goal $p(\mathbf{a}|t^*, \mathbf{o}^*)$. And then find the most probable configuration. However, this method may suffer from the inaccuracy posed in the defuzzification step as explained in Section II-C. A small error in the reconstruction of finger configuration may result in unstable grasps or even impossible situations where fingers penatrate the object surface. A better way is to have a set of preplanned grasp hypotheses that are parameterized by $\mathbf{a}$ (see the four example hypotheses in Fig. 6). The robot can then use $p(\mathbf{a}|t^*, \mathbf{o}^*)$ to select the most suitable one in order to pour with the selected mug. Notice here the optimal grasping policies (including selected object and grasp) may be different from the exact human demonstration. However, this is acceptable as long as the task goal is satisfied and the action is compatible with the embodiment of the robot.

V. Conclusion

We have presented a method for task/goal prediction within the scope of goal-directed grasp imitation. The method allows a robot to predict human intention during object grasping based on a novel hand-object 3D tracking framework and a probabilistic modeling approach that encode the grasping tasks. The tracking framework is able to extract hand-object grasp configurations using Kinect-based natural observation on human demonstrations. This observation is the basis for the goal inference using Bayesian networks which encode the embodiment-specific relations between task semantics and grasping strategies for human demonstrators. Real-world related problems, like noise in observations, high dimensionality and complex probability distribution, are tackled through a SOM-powered soft discretization technique. The learned Bayesian networks represent the internal models for visual-manual tasks similar to [24]. But far beyond the scope of [24], our method generalizes to multiple hands, objects and tasks. The generative modeling approach is able to deal with uncertain, incomplete perception in real world applications. Interestingly, when the

models are learned from simulated data, they were potent enough to provide high accuracy in task prediction over real-world data. We also pointed out the method's place within a full goal-directed imitation framework by establishing the method's connection to actual robot execution.

In this paper, we used a SOM-based method to discretize high-dimensional continuous data. A similar problem has been addressed by [31] where the authors proposed a multivariate discretization approach based on latent variable models. Since the discretization is not the focus of this paper and due to space limitation, we did not evaluate our method against that in [31]. This will be done in the future work.

In addition, the current Bayesian network is a static model encoding task information only at the moment of final grasp configuration. However, the dynamic manipulation of object after lifting contains rich information about the task. Especially for some tasks (e.g., pouring and drinking) that are difficult to differenciate by only the static grasp configuration, sequential data of object manipulation will be much more powerful to solve the ambiguity. We are now investigating how to exploit this by modeling the sequence of grasping and manipulation actions using dynamic Bayesian networks. The model should be used for both task recognition during human demostration and action reproduction on a robot platform.

ACKNOWLEDGMENTS

This work was supported by the IST-FP7-IP-288533 project robohow.cog and Swedish Foundation for Strategic Research.

REFERENCES

[1] A. A. Argyros and M. Lourakis, "Real-time Tracking of Multiple Skin-colored Objects with a Possibly Moving Camera," in *ECCV*. Springer, 2004, pp. 368–379.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Springer Handbook of Robotics*, 2008, pp. 1371–1394.

[3] C. Chow and C. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[4] I. Ebert-Uphoff, "A Probability-Based Approach to Soft Discretization for Bayesian Networks," Georgia Institute of Technology. School of Mechanical Engineering, Tech. Rep., 2009.

[5] S. Ekvall and D. Kragic, "Grasp Recognition for Programming by Demonstration," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Jan. 2005, pp. 748–753.

[6] L. D. Fu and I. Tsamardinos, "A Comparison of Bayesian Network Learning Algorithms from Continuous Data," in *AMIA*, 2005.

[7] J. G. Greeno, "Gibson's Affordances," *Psychological Review*, vol. 101, no. 2, pp. 336–342, 1994.

[8] D. B. Grimes and R. P. N. Rao, "Learning Actions through Imitation and Exploration: Towards Humanoid Robots that Learn from Humans," in *Creating Brain-Like Intelligence*, ser. Lecture Notes in Computer Science, vol. 5436. Springer, 2009, pp. 103–138.

[9] C. Huang and A. Darwiche, "Inference in Belief Networks: A Procedural Guide," *Int. Journal of Approximate Reasoning*, vol. 15, pp. 225–263, 1994.

[10] K. Huebner, "BADGr – A toolbox for box-based approximation, decomposition and GRasping," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 367–376, 2012.

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

[12] H. Kjellström, J. Romero, and D. Kragić, "Visual object-action recognition: Inferring object affordances from human demonstration," *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 81–90, 2011.

[13] T. Kohonen, *Self-Organizing Maps*. Berlin/Heidelberg: Germany: Springer, 1995, vol. 30.

[14] W. V. Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy Sets and Systems*, vol. 108, no. 2, pp. 159–178, 1999.

[15] P. Leray and O. Francois, "BNT Structure Learning Package: Documentation and Experiments," Université de Rouen, Tech. Rep., 2006.

[16] C.-y. Lin, J.-x. Yin, L.-h. Ma, and J.-y. Chen, "An Intelligent Model Based on Fuzzy Bayesian Networks to Predict Astrocytoma Malignant Degree," in *2006 IEEE Conference on Cybernetics and Intelligent Systems*. IEEE, Nov. 2006, pp. 1–5.

[17] J. Maycock, J. Steffen, R. Haschke, and H. J. Ritter, "Robust tracking of human hand postures for robot teaching," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco: IEEE, 2011, pp. 2947–2952.

[18] Microsoft Corp. Redmond WA, "Kinect for Xbox 360."

[19] A. T. Miller and P. K. Allen, "GraspIt! A Versatile Simulator for Robotic Grasping," *IEEE Robotics and Automation Magazine*, 2004.

[20] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning Object Affordances: From Sensory–Motor Coordination to Imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[21] C. L. Nehaniv and K. Dautenhahn, *The Correspondence Problem*. MIT Press, 2002, pp. 41–61.

[22] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect," in *BMVC*, 2011.

[23] ——, "Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *ICCV*, 2011.

[24] E. Oztop, D. Wolpert, and M. Kawato, "Mental State Inference using Visual Control Parameters," *Cognitive Brain Research*, vol. 22, no. 2, pp. 129–151, February 2005.

[25] C. Papazov and D. Burschka, "An efficient ransac for 3d object recognition in noisy and occluded scenes," in *Proceedings of the 10th Asian Conference on Computer Vision (ACCV'10)*, 2010.

[26] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3D Geometry Matching for Grasping of Known Objects in Cluttered Scenes," *International Journal of Robotics Research*, vol. 31, no. 4, pp. 538–553, 2012.

[27] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009.

[28] R. Rao, A. Shon, and A. Meltzoff, "A Bayesian Model of Imitation in Infants and Robots," in *Imitation and Social Learning in Robots, Humans, and Animals*, 2004, pp. 217–247.

[29] G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[30] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Embodiment-Specific Representation of Robot Grasping Using Graphical Models and Latent-space Discretization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011.

[31] ——, "Multivariate discretization for bayesian network structure learning in robot grasping," in *IEEE Int. Conf. on Robotics and Automation*, 2011.

[32] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning Task Constraints for Robot Grasping using Graphical Models," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010.

[33] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele, "Functional Object Class Detection Based on Learned Affordance Cues," in *Computer Vision Systems*, ser. Lecture Notes in Computer Science, A. Gasteratos, M. Vincze, and J. K. Tsotsos, Eds. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2008, vol. 5008, ch. 42, pp. 435–444.

[34] D. Verma and R. Rao, "Goal-based imitation as probabilistic inference over graphical models," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, pp. 1393–1400.

[35] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map." *IEEE transactions on neural networks*, vol. 11, no. 3, pp. 586–600, 2000.