Utilizing Gaze Behavior for Inferring Task Transitions Using Abstract Hidden Markov Models

Lord Kenneth Pinpin¹ , Daniel Fernando Tello Gamarra², Roland Johansson³, Cecilia Laschi⁴ and Paolo Dario⁴

¹Standard Insurance Co. Inc.
Kenneth.pinpin@issico.com

²Universidade Federal de Santa Maria (UFSM)
Control and Automation Engineering, Santa Maria, Rio Grande do Sul, Brazil.
daniel.gamarra@ufsm.br

³Umeå University, Umeå, Sweden.
roland.s.johansson@umu.se

⁴Scuola Superiore Sant'Anna
ARTS Lab(Advanced Robotics Technology and Systems Laboratory), Pisa, Italy.
{cecilia.laschi, paolo.dario}@sssup.it

Abstract We demonstrate an improved method for utilizing observed gaze behavior and show that it is useful in inferring hand movement intent during goal directed tasks. The task dynamics and the relationship between hand and gaze behavior are learned using an Abstract Hidden Markov Model (AHMM). We show that the predicted hand movement transitions occur consistently earlier in AHMM models with gaze than those models that do not include gaze observations.

Keywords: Gaze Modeling; Gaze Tracking; Probabilistic Graphical Model, AHM, Task Transitions Prediction.

1 Introduction

Robotic assistive devices are envisioned to assist humans in scenarios where the person has reduced physical ability. For example, a hybrid bionic system such as a hand/arm exoskeleton may be used to augment muscular strength and aid in manual dexterity. In such a scenario, giving the assistive system the ability to predict and anticipate human hand movements would be an advantage for ensuring safety and providing a natural interface for using the device.

In this paper, we demonstrate that observed human gaze behavior can be proactively used to improve the prediction of hand movement transitions while doing simple, well-learned object manipulation tasks. Our earlier effort [22] used a simple method of tracking gaze fixations as a way of triggering pre-defined robot movements that corresponded to task segments. This method relied on the key observation of human behavior that gaze fixations marked spatial goals of successive action phases while performing well-learned object manipulation tasks [10]. Gaze fixations in landmarks where contact is about to be made were found to be obligatory while those in possible contact points such as obstacles were optional. These gaze fixation areas reveal "visual control points" for goal completion of each action phase. Thus, monitoring gaze fixation behavior can reveal task segmentation and is useful for hand motion prediction in the context of a task.

Though simple, the method of monitoring when gaze fixations occur at specific landmarks was shown to reliably determine where the next contact is about to be made [22]. However, aside from not being able to predict

the exact time of task transition, several other shortcomings limit the usefulness of this approach. First, several key components of the method had to be manually set up: the gaze fixation areas and gaze velocity threshold needed to be predetermined instead of being discovered automatically. Second, the state-machine that describes the task transitions was pre-programmed. Third, noise-filtering was very limited. Also, there was no mechanism for continuous optimization or learning.

We address these problems by adopting a probabilistic machine learning approach. In particular, we use an Abstract Hidden Markov Model (AHMM) for modeling the manipulation tasks and the corresponding gaze behavior. First introduced by Bui [4], [5], the AHMM is a hierarchical statistical model useful for representing activities and was used for inferring what policy the agent is following by observing its behavior and effects on the world. As Murphy demonstrated in [24], a 1-level AHMM can be represented as a Dynamic Bayesian Network (DBN). Murphy also demonstrated how the model parameters of the AHMM could be learned from data.

We also use a 1-level AHMM represented as DBN to model recorded task sequences and gaze behavior and use Expectation Maximization (EM) [5] for learning the model parameters from recorded data. To infer the discrete hidden variables that represented the goals and states of the task being observed, we used offline-smoothing. The position of the object and the gaze fixation locations are represented as conditional probability distributions, which are incorporated in the structure of the AHMM as discrete hidden state variables. Thus the main contributions of this paper are fourfold. First, we are able to determine, in an unsupervised manner, the relevant location and size of gaze fixation areas, the gaze transitions, and object positions and task transitions. Second, is the use of Dynamic Bayesian Networks for modeling gaze. Third, by using AHMM, a probabilistic model, we encode the relationship between gaze behavior and task. Fourth, we showed that incorporating gaze improves the prediction of task transitions.

1.1 Related Work on Gaze Modeling

Human gaze behavior has been extensively studied and falls into two general categories: bottom-up saliency studies that hypothesize that certain scene features naturally attract gaze (which are represented as saliency maps) [26] and are typically found in attention models [28] [33], biologically-inspired object detection/search mechanisms [17], [36], [35], and saliency search modulation [28]; and top-down cognitive control approaches that emphasize the role of behavioral objectives in determining gaze or eye movement. Task based models of gaze seek to describe how the demands of a task or of competing tasks [32] influences the gaze strategy. Hayhoe [14] and Land [20] provide comprehensive reviews of eye-tracking studies done in the context of natural behaviors. These studies point out clearly that gaze is heavily linked to the ongoing demands of the executed task. Also Gonzelez in [13] proposes a mathematical model to explain how head direction affects the perception of gaze direction.

Robot related applications of tracking gaze direction has been applied to multi-modal communication [21], testing neurophysiological control models [9] and endowing robots with human-like gaze behavior [15], [31] and [16], Kishore in [30] explored how a full-body ownership illusion can be induced over a remote humanoid robot. However, little research has been done in fusing observed human gaze with observed hand movements for obtaining better predictions of task transitions or task objectives. While Rothkoph et al [29] do emphasize the role of the contextual demands of the task, their main objective is to better predict the direction of gaze by knowing the phase of the action sequence (task-dependent) than by local features of the visual scene (bottom up). In contrast, our objective is to utilize the observed gaze behavior to obtain better estimates of the transitions of the underlying task. This is one of the key contributions of this paper.

The rest of the paper is organized as follows. In section 2 we describe the experimental setup followed by a brief presentation of the probabilistic modeling tools used in Section 3. Our results are presented and discussed in Section 4. In the final section, we draw some conclusions and give an outline of future research directions.

2 Experimental Setup

The data we used for training the AHMM models were from the recorded trials in [22] and it would be necessary to format the data to make it publicly available. We summarize here the experimental setup used to collect that data. The basic setup is illustrated in Figure 1 that depicts a participant doing the experiment in the equipment and a software plot corresponding to the gaze tracker, the subject can move its head freely. Figure 2 helps to describe better the experiments where a subject is seated in front of the manipulation area holding the bar (2 x 2 x 8 cm) and asked to make contact to two target points in particular order before returning to the home position (contact position 0). The location of the contact points and the dimension of the work area are identical to [19]. All movements take place on a vertical frontal plane, termed the work plane, which is located between the subject and the faceLAB gaze-tracking hardware [1].

Figure 1: Basic experimental setup with a participant involved in the experiments.

Figure 2: Task movement primitives, contact points, and gaze landmark zones (yellow squares) of the gaze experimental setup of [1].

2.1 Gaze Tracking

The faceLAB system was configured for precision mode which allowed head tracking and individual eye tracking at 60 Hz. Although the faceLAB system tracks many head and gaze variables, for the purposes of this experiment, we used the gaze fixation point which is computed automatically by faceLAB as the intersection between the unified gaze ray and the work plane, defined relative to the world coordinate frame. Left and right eyeball gaze angles are also tracked to compute the gaze angular velocity. In addition, the software performed real-time visualization, data recording, and gaze and video playback (see [22] for software implementation details).

2.2 Tasks

Each subject is asked to perform four tasks shown in Figure 3. There are four goal-directed tasks performed by the user. In the task 1: the subject moves the bar from home position to target 2, back to home, home to target 1, back to home (repeat 4 times); task 2: the subject moves the bar from home position to target 2, then to target 1, then back to home position (repeat 4 times); task 3: the same as Task 2 but count 5 seconds at each contact position; and finally, in task 4: the subject will make 30 contacts in any order (consecutive contacts should be at

different target positions).

Tasks 1 to 3 are performed with 4 repetitions. Task 4 is performed for a total of 30 contacts with the order of the contacts determined by the subject. Tasks 1, 2, and 3 may be considered examples of well-learned tasks that require little cognitive processing to accomplish the required steps. Tasks 1, 2, and 3 are simple enough to require only a decision to follow the predefined instructions to fulfill the task. In contrast, Task 4 evidently requires more decision making on the part of the subject since there is no predefined sequence of contacts that must be performed. In addition, the subject must count the number of contacts made.

Figure 3: Four goal-directed tasks performed by the user. Task 1: subject moves the bar from home position to target 2, back to home, home to target 1, back to home (repeat 4 times). Task 2: subject moves the bar from home position to target 2, then to target 1, then back to home position (repeat 4 times). Task 3: same as Task 2 but count 5 seconds at each contact position. Task 4: make 30 contacts in any order (consecutive contacts should be at different target positions).

Task 3, though similar in spatial sequence with Task 2 is different in that it also demands a temporal compliance to fulfill the task properly. For the intended future robotic applications that would include gaze, fulfilling spatial and temporal requirements are equally important. So Task 3 is Task 2 with a different temporal requirement.

2.3 General Procedure

Four right-handed male subjects were asked to perform the four tasks. A faceLAB forward-only head model was then made for each subject. Before the beginning of each task, the subject is given instructions on what to do with the object. Each was instructed to keep their head stationary while doing the task since the viewable area in precision faceLAB mode is limited. With the exception of Task 3, no time limit was given for each segment of the task. The only additional instruction and information given to each subject before performing the tasks was to move the object in a natural manner.

2.4 Object Color-Tracker

Aside from the gaze variables tracked by faceLAB, the position of the bar held by the subject is recorded using a video camera. A probabilistic color-based tracker was used for monitoring the object position in the video image (24 fps at 640x480 pixels). The tracker relies on the principle of comparing the color content of candidate regions to a reference color histogram but embedded in a sequential Monte Carlo framework. This requires a color

likelihood model based on color histogram distances, a dynamic state-space model, and sequential approximation of the resulting posterior distribution with a particle filter.

We adopt the color likelihood model and the dynamic state-space model introduced by Nummiaro [27] but replace the particle filter with an Unscented Particle Filter (UPF) that is known to have better performance [7] and is better able to deal with degeneracy of the samples (particles). Similar use of the UPF for object tracking was reported in [34] and [18]. We refer the interested reader to these references for details on the use of the UPF for vision based object tracking. The Color-tracker was used in real time.

2.5 Task Transition Ground Truth

For purposes of comparing the models task transition prediction ability we obtain temporal reference data from the video of the subjects performing the tasks. The video at which the subject hand starts to move away from a contact point is the defined to be the task transition ground truth. This is done by performing a frame-by-frame visual inspection of the video.

3 Theoretical Background

3.1 Hidden Markov Models (HMM)

A Hidden Markov Model is a stochastic finite automaton, where each state generates an observation. Where X denotes the hidden state and Y_t to denote the observation. The parameters of the model are the initial state distribution, $\pi(i) = P(X_i = i)$, the transient model, $A(i, j) = P(X_t = j | X_{t-1} = i)$, and the observation model $P(X_t | Y_t)$. $\pi(\cdot)$ represents a multinomial distribution. The transition model is usually characterized by a conditional multinomial distribution: $A(i, j) = P(X_{t=j} | X_{t-1=i})$, where A is a stochastic matrix.

3.2 Dynamic Bayesian Network

Probabilistic networks, also known as Bayesian networks are a specific type of graphical model that is represented as a Directed Acyclic Graph (DAG) (definition from state of the art BN). Nodes in DAG are graphical representation of objects and events that exists in the real world, and are usually termed variables or states. Causal relations between nodes are represented drawing an arc (edge) between them. If there is a causal relationship between the variables, the edge will be directional, leading from the cause variable (parent) to the effect variable (child). For each variable in the DAG there is probability distribution function (pdf), which dimensions and definition depends on the edge leading into the variable. In a DAG each variable (node) has at most one parent.

A dynamic Bayesian network is a way to extend Bayesian networks, to model probability distributions over semi-infinite collection of random variables $Z_1, Z_2, \dots$. The variables will be divided into $Z_t = (U_t, X_t, Y_t)$, that represents the input, hidden and output variables of a state-space model. A dynamic Bayesian network is defined to be a pair, $(B_1, B_{\rightarrow})$, where B_1 is a Bayesian network which defines the prior $P(Z_1)$, and $B_{\rightarrow}$ is a two-slice temporal Bayes network (2TBN), the nodes in the first slice of a 2TBN do not have any parents associated with them, but each node in the second slice of the 2TBN has an associated conditional distribution (CPD) which defines $P(Z_t^i | P_a(Z_t^i))$ for all $t > 1$, a 2TBN defines $P(Z_t | Z_{t-1})$ by means of DAG as follows:

$$P(Z_t, Z_{t-1}) = \prod P(Z_t^i | P_a(Z_t^i)) \quad (1)$$

Where Z_t^i is the i th node at time t , which could be a component of X_t, Y_t or U_t , and $P_a(Z_t^i)$ are the parents of Z_t^i in the graph.

3.3 Representing Hidden Markov Models and their variants as Dynamic Bayesian Networks

It is possible to represent an HMM as a DBN as shown in Figure 4. The shaded nodes are the observed variables and the clear nodes represent hidden variables. In Bayesian Networks, we must define the conditional probability distribution (CPD) of each node given its parents. For the HMM in Figure 4, this means defining $P(X_t)$, $P(X_t | X_{t-1})$ and $P(Y_t | X_t)$. The CPD for $P(X_1)$ is usually represented as a vector, the CPD for $P(X_t | X_{t-1})$ is usually represented as a stochastic matrix, the CPD for $P(Y_t | X_t)$ can take different forms, if Y_t is discrete, we could use a conditional multinomial, represented as a stochastic matrix. If Y_t is continuous, we could use a conditional Gaussian or a conditional mixture of Gaussians.

Figure 4: An HMM represented as an instance of a DBN, unrolled for three slices.

3.4 Abstract Hidden Markov Model

An AHMM allows us to decompose a behavior into hierarchal sub-tasks/sub-behaviors. Following the terminology of [3], a policy is a probabilistic mapping from states to actions: $\sigma_\pi(s, a) = P(\text{do action } a | \text{in state } s)$. An abstract policy is a probabilistic mapping from states to lower level abstract policies, or "macro actions". An abstract policy can call a sub-policy, which runs until termination, returning control to the parent policy; the sub-policy can in turn call lower-level abstract policies, until we reach the bottom level of the hierarchy, where a policy can only produce concrete actions.

3.4.1 1-Level AHMM

The simplest AHMM is the model shown in Figure 5, where $Q_t^1 = S_t$ which represents the goal an agent is currently following and $Q_t^2 = S_t$ is the state of the world. This can be used to model the situation where an agent follows a plan made of an ordered sequence of subgoals $(g_1, \dots, g_n)$ and that Y_t is the observation of the current state S_t . The transition or evolution of the state depends on the current state and the current goal. Graphically, this is represented by the links from the current goal G_t and the previous state S_{t-1} to the current state S_t . Therefore, the conditional probability transition matrix is:

$$P(S_t = j \mid G_{t-1} = g, S_{t-1} = i) = A_g^S(i, j) \quad (2)$$

During plan execution, the current goal is assumed to stay the same until it has been achieved at the state level. The node/variable F is a binary variable that represents whether the current goal G_t is achieved. This is represented by the links from S_t and G_t to F_t^G . Given these conditioning links, the transition matrix for is F_t^G is:

$$P(F_t^G = 1 \mid G_t = g, S_{t-1} = i) = A_g^S(i, \text{end}) \quad (3)$$

If F_t is true, we assume that in the next time instant, the current goal will be replaced, otherwise the current goal will persist:

$$P(G_t = j \mid G_{t-1} = i, F_t^G = f) = \begin{cases} \delta(i, j) & \text{if } f = 0 \\ A_G(i, j) & \text{if } f = 1 \end{cases} \quad (4)$$

Where $A^G(i, j)$ could be a stochastic transition matrix or a deterministic successor-state function for goals.

3.5 Inference in AHMM

Given the sequence of observations, the goal of inference is to estimate the hidden variables in the model. For example, if the observations Y_t are the object positions $o_i(x_i, y_i)$ in pixels detected by our color based tracker, we can use the AHMM model to compute the S_t or G_t using an inference algorithm. There are three possible kinds of inference possible with probabilistic graphical models such as the AHMM: filtering, smoothing and prediction. The Filtering is the most common inference problem in online analysis which is done by recursively estimating the belief state X_t using Bayes rule:

$$P(X_t | Y_{1:t}) \propto P(y_t \mid X_t, y_{1:t-1}) P(X_t \mid y_{t-1}) = P(y_t \mid X_t) \left[\sum_{x_{t-1}} P(X_{t-1} \mid y_{1:t-1}) \right] \quad (5)$$

Figure 5: A 1-level AHMM. turns on when the state satisfies the current goal. However, since there is no arc from F_t^G to S_t to reset the lower model, it may simply depend on the previous state.

Since y_t is only dependent on X_t (Markov assumption). Filtering involves predicting the $P(X_t|y_{1:t-1})$ from the prior belief state $P(X_{t-1}|y_{1:t-1})$ and updating it using the latest observation y_t to obtain the posterior belief state.

Smoothing is the process of computing the value of the state in the past, given all the observations up to the current time. That is, $P(X_{t-lag}|y_{1:t})$ this is called 'fixed-lag smoothing'. In the offline case, called fixed-interval smoothing, this means computing $P(X_t|y_{1:t})$ for all $1 \leq t \leq T$.

In prediction we can estimate future states given evidence available at the current time step, $P(X_{t+h} | y_{1:t})$ where $h > 0$ is how far ahead we want to predict. For our experiments we used offline smoothing using Kevin Murphy's Interface Algorithm [24], also known as the dynamic junction-tree algorithm, which is more efficient than the Frontier Algorithm [37] for exact inference in DBNs. The dynamic junction-tree algorithm uses the static junction-tree algorithm as a subroutine, however. Details on the algorithm and how the efficiencies were achieved are beyond the scope of this paper. Interested readers are encouraged to read Murphy [24].

For large AHMMs and for very long sequences, exact inference becomes impractical. Approximate inference methods such as particle filtering (PF) [8] or Rao-Blackwellized Particle Filtering (RBPF) can be used for fast and efficient inference. In [5], Bui showed how RBPF can be used inference in an AHMM.

3.6 Learning in AHMM

The parameters in the AHMM need to be learned from data before inference can be done. We use Expectation-Maximization (EM) to learn the parameters of the AHMM. EM is an iterative technique for finding maximum-likelihood estimates of parameters in probabilistic models, where the model depends on hidden/unobserved variables [6]. EM alternates between performing an expectation (E) step, which computes an expectation of the log-likelihood by including the latent variables as if they were observed, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step.

For AHMM, during the E-step exact inference methods such as the dynamic junction-tree algorithm can be used to compute the required expected sufficient statistics of the hidden nodes, given the observed data. The M-step then treats these supplied values as if they were observed and re-estimates the model parameters

3.7 AHMM Training setup

For each subject and for each recorded task we created two types of models: one without gaze and one with gaze. For the Tasks 1, 2, and 3, the entire sequence of 4 repetitions was used as the training data for the EM. The length of each sequence varied from 800 to 1100 time steps. The EM training of the model parameters and subsequent offline smoothing were done using the Bayes Net Toolbox for Matlab [23].

3.7.1 Model Definition

We used 1-level AHMM for all the modeling experiments. Therefore, it is assumed that the number of high level goals and the levels of the hierarchy are known instead of being learned from data. Since there are 3 contact points in the task environment, for all models, we fixed to 3 the number of discrete values that hidden node G could take. For all models, we fixed to 10 the number of possible discrete values of hidden node S, which represents

the discrete quantization of the actual observed values (object motion as seen from the video camera, the gaze fixation point on the work plane, and the gaze velocity). The observed variable Y takes continuous values and modeled as a multivariate Gaussian. Training Data for AHMM.

For models without gaze, we selected as the observed values the x-y position (in pixels) and x-y velocity (in pixels/sec) of the object in the video frame obtained by the UPF color-tracker. So in effect, the observed variable Y is represented as a 4-dimensional multivariate Gaussian.

For models with gaze, in addition to the position and velocity of the object, the gaze fixation point (x, y location in the work plane) and the gaze velocity in rad/sec was included as part of the observed values. Effectively, for these models the variable Y is represented as a 7-dimensional multivariate Gaussian.

We trained the models in an unsupervised manner. That is, the highest level composed of the G nodes was unobserved. Part of the motivation to train the models in this manner was to test whether useful models can still result without undue assistance from labeling the high level goals that the sequences of observations represent. Also, we believe that in practice any learning system must automatically learn high level goals from data.

We also tried training with combined data from Tasks 1, 2, and 3 (for each subject) and used the resulting model in inferring task transitions for Task 4 (of each subject). Two models were created in this manner: one without gaze and one with gaze observed values.

4 Results

Though EM guarantees that the likelihood will not decrease with each iteration, it does not guarantee convergence to the maximum likelihood. So we ran EM several times (6 times) for each recorded task data as a single sequence of observations and selected models that resulted in highest log-likelihood. We examine the results and in following sections show the learned observation distributions, compare the inferred task transitions of models with gaze and models without gaze to ground truth, and the relationship between gaze zones and tracked object positions.

4.1 Inferred Task Transitions

To infer task transitions we inspect the posterior distribution which is shown in Annexo A for Tasks 1, 2 and 3, for AHMM models that observe only the object position and velocity. The corresponding learned object positions for each of the 3 models of the tasks shown in Appendix A is depicted in Figure 6(a).

4.2 Inferring Task Transitions with Gaze Observations

Annexo B shows the posterior distribution $P(S_{0:t}|Y1:t)$ inferred from observed data of Tasks 1, 2, and 3 (from the same subject) with models that included the gaze values as part of the observation variable Yt . Compared with models that did not have gaze variables the exit transitions from the key locations 1, 2, 3 occur a few frames earlier. When compared with the ground truth data, the inferred positions indicate that the transitions are predicted in advance by a few frames, with one notable exception in Task 2, the exit transition from position 2 is delayed compared to ground truth (see Figure 9(b)). The corresponding learned object positions for each of the 3 models of Appendix B is shown in Figure 6(b).

4.3 Gaze and Task Dynamics

The EM-trained AHMM models encoded conditional transition matrices and spatial distributions that were learned from data instead of being manually set. Using the AHMM model for task 3 as an example, the encoded relationship between object position and gaze within the model can be illustrated by viewing the posterior distribution of the hidden state variable, $P(S_{0:t}|Y1:t)$, in conjunction with the learned observation distributions for gaze and object position $P(Y_t|S_t)$. For example, in Figure 7, using the inferred value of the hidden state variable S_t as an index and examining each column for the most probable value of S_t , we obtain the following state sequence: 10, 1, ..., 1, 4, 2, ..., 2, 6, 8, 3, ..., 3, 9, 10, 1. Referring to the gaze and object distributions, this means when $S_t = 10$ the subject briefly looked where he was holding the object, (1) fixated the end of the bar, (4) fixated the contact point 2 as he started moving the bar, (2) fixated the contact point while the bar touched it, (6) fixated the obstacle while the object is still in current contact point and then (8) fixated contact point 1 in advance of the object, which around this time, is navigating the obstacle, (3) he watches while contact is made at contact point 1, (9) gaze shifts to the home position shortly after the object leaves the contact point 1, and (10) as the object approaches the home position the gaze shifts towards the end of the object (1), completing one cycle.

(a) Learned object positions from models in Appendix A. (a) Task 1 learned object positions corresponding to numbered positions for the model used in Appendix A-Figure 9(a). Note that positions 1 and 4 are overlapping. (b) Task 2 learned object positions corresponding to numbered positions for the model used in Appendix A-Figure 9(b). (c) Task 3 learned object positions corresponding to numbered positions for the model used in Appendix A-Figure 10. Note that positions 3 and 10 are overlapping.

(b) Learned object positions from models in Figure 10. (a) Task 1 learned object positions corresponding to numbered positions for the model used in 10(a). Note that positions 1 and 4 are overlapping. (b) Task 2 learned object positions corresponding to numbered positions for the model used in Figure 10(b). (c) Task 3 learned object positions corresponding to numbered positions for the model used in Figure 10(c). Note that positions 3 and 10 are overlapping.

Figure 6: Learned object positions from models without gaze and with gaze.

Figure 7: Gaze and task dynamics for Task 3. (Top) Inferred values of hidden state variable S_t . (Middle) Learned object positions. (Bottom) Learned gaze fixation zones.

4.4 Testing AHMM Models with Task 4

We compare two models trained with the combined data sequences from tasks 1, 2, and 3 one without gaze and one with gaze observations. The trained models were tested with the data from Task 4 to determine the usefulness for predicting unknown task sequences. The results are shown in Figure 8. The AHMM model trained without gaze was able to track the task transitions remarkably well. However, without the additional information provided by gaze the transitions are delayed by a few frames when compared with ground truth. In comparison the AHMM model trained with gaze observations estimated states with slightly larger covariances than the model trained without gaze observations (see Figure 7 with bigger ellipses representing the state estimation covariances). This may be due to insufficient data for training, since the gaze-based models have more parameters to learn. However, we note that its predicted task transitions occur earlier compared to ground truth. This suggests that models that utilize gaze are more useful for predicting task transitions at the expense of model complexity, position estimation accuracy and the need for larger training data. Hence, there is a trade-off between object position and task transition estimation accuracy.

5 Discussion

The results show that task-based models of gaze can be learned from observed behavior using the AHMM. These results bolster previous studies on gaze behavior in relation to task by providing a practical way of creating

(a) AHMM model trained with data sequences from Tasks 1, 2, and 3 (no gaze observations). Left: Learned objects distributions. Right: posterior distribution $P(S_{0:t}|Y_{1:t})$ for task 4 (showing only first 700 of 2100 frames). White vertical bars represent ground truth that indicates when object begins to move away from a contact point.

(b) AHMM model trained with data sequences from Tasks 1, 2, and 3 (with gaze observations). Left: Learned objects distributions. Right: posterior distribution $P(S_{0:t}|Y_{1:t})$ for task 4 (showing only first 700 of 2100 frames). White vertical bars represent ground truth that indicates when object begins to move away from a contact point.

Figure 8: AHMM model trained without gaze and with gaze.

probabilistic models that can be used not only for analysis of human behavior but also for integration with human assistive systems.

Compared with [22], the probabilistic approach taken here improves on key areas such as noise filtering, learning from data, and task transition prediction. Unlike the ad-hoc approach taken in [22] where noisy gaze data was replaced with older data, inference in DBNs such as the AHMM is more robust in estimating states given noisy observations. Model parameter learning via EM eliminated the need to hand code the state-machine that describes the task dynamics this was encoded in the learned conditional transition tables of the hidden variables of the model. Similarly, instead of presetting the gaze fixation areas manually, the dimensions and locations of the gaze fixations were simultaneously learned from recorded observations and encoded in the mean

and covariance matrices of the multivariate Gaussian distributions of the observation variable. With regards to predicting task transitions, inference with the AHMM models with gaze, when trained with sufficient examples, provided a means to anticipate transitions between task segments. This is certainly an advantage for systems that monitor the human state for purposes of ensuring safety while providing task assistance.

6 Conclusion

Previous work on gaze behavior has shown clearly that gaze behavior is influenced by the demands of the task being performed. Our contribution to the study of human gaze field is four-fold. First, we are able to determine, in an unsupervised manner, the relevant location and size of gaze fixation areas, the gaze transitions, and object positions and task transitions - rectifying many of the limitations of the gaze interface developed in [22]. Second, is the use of Dynamic Bayesian Networks for modeling gaze. Third, by using AHMM, a probabilistic model, we encode the relationship between gaze behavior and task. Fourth, we show that probabilistic inference on these models can provide early prediction of task transitions.

Although we have shown that gaze can be useful for improving task prediction there is much room for improvement. Clearly, we are modeling very simple tasks and have limited movements to a 2D plane. Also, we have assumed that the number of high level goals and the levels of the hierarchy are known. Ideally, model structure and topology should be learned from the data in addition to the model parameters. This is still an open research area in machine learning, but some success has been demonstrated in online estimation of both model structure and parameters such as [2]. For the future we intend to explore other DBN structures such as the Switching Kalman Filter or the extension of the AHMM, the Abstract Hidden Markov Memory Model (AHMEM) [25], and replace EM with Variational Bayes [3] for model selection and use approximate inference methods such as Rao-Blackwellized Particle Filtering (RBPF). We intend to use such tools to help us create task-based models of human gaze that can be useful in human assistive systems, human-robot interaction scenarios or used the same probabilistic approach to construct models in other sensory-motor tasks in robotics [12] and [11].

Acknowledgements

This work is supported by the ROBOTCUB project (*IST – 2004 – 004370*), funded by the European Commission through Unit *E5* Cognitive SYSTEMS. Also we would like to thank the IIT (Italian Institute of Technology) for its financial support.

References

- [1] Facelab face and eye tracking system. <http://www.seeingmachines.com>.
- [2] C. Andrieu, N. De Freitas, and A. Doucet. Sequential mcmc for bayesian model selection. In *Proceeding of the IEEE Signal Processing Workshop on Higher Order Statistics*, pages 130–134, 1999.
- [3] M.J. Beal. *Variational Algorithms for Approximate Bayesian Networks*. PhD thesis, PhD Thesis Gatsby Computational Neuroscience, University College London, 2003.
- [4] H.H. Bui. Policy recognition in the abstract hidden markov models. *Journal of Artificial Intelligence Research*, 17:451–493, 2003.
- [5] H.H. Bui, S.Venkatesh, and G.West. *Hidden Markov Models: applications in computer vision*, chapter Tracking and Surveillance in wide-area spatial environments using the abstract hidden Markov model.
- [6] A.P. Dempster and N.M. Laird and D.R. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Serie B (Methodological)*, 39:1–38, 1977.
- [7] R. Van der Merwe, N. De Freitas, A. Doucet, and E.A.Wan. The unscented particle filter, 2000.
- [8] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, London-UK, 2001.
- [9] E.S.Maini, L.Manfredi, C.Laschi, and P.Dario. Bioinspired velocity control of gaze shifts on a robotic antropomorphic head. *Autonomous Robots*, 25:37–58, 2008.
- [10] B. Fritzke, G. Tesauero, D.S. Touretsky, and T.K.Leen. *Advances in Neural Information Processing Systems*, chapter A Growing Neural Gas Network Learns Topologies.
- [11] D.F.T. Gamarra. Ballistic reaching using visuomotor maps constructed with som neural networks for a robot manipulator. *International Review on Modeling and Simulations*, 23:1926–1930, 2014.

- [12] D.F.T. Gamarra and M.A.S.L. Cuadros. Forward models for following a moving target with the puma 560 robot manipulator. *Iberoamerican Journal of Artificial Intelligence*, 18:31–48, 2015.
- [13] M. Gonzalez-Franco and P. A. Chou. Non-linear modeling of eye gaze perception as a function of gaze and head direction. In *Int'l Workshop on Quality of Multimedia Experience (QoMEX)*, September 2014.
- [14] M. Hayhoe and D. Ballard. Eye movements in natural behavior. *Trends in Cognitive Sciences*, 9:188–194, 2005.
- [15] M.W. Hoffman, D.B. Grimes, A.P. Shon, and R.P.N. Rao. A probabilistic model of gaze imitation and shared attention. *Neural Networks*, 19:299–310, 2006.
- [16] I. Goga and A. Billard. Attention mechanisms for the imitation of goal directed action in developmental robots. In *6th International Conference in Epigenetic Robotics*, pages 1–20, 2006.
- [17] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506, 2000.
- [18] J. Li, H. Yu, L. Zhou, H. Liang, and L. Wang. An adaptive unscented particle filter tracking algorithm based on color distribution and wavelet moment. In *2nd IEEE Conference on Industrial Electronics and Applications, ICIEA*, 2007.
- [19] R.S. Johansson, G. Westling, A. Backstrom, and J.R. Flanagan. Eye-hand coordination in object manipulation. *Journal of Neuroscience*, 21:6917–6932, 2001.
- [20] M.F. Land. Eye movements and the control of actions in everyday life. *Prog Retin Eye Res*, 25:37–58, 2006.
- [21] L. Chen. Incorporating gesture and gaze into multimodal models of human-to-human communication. pages 211–214, 2006.
- [22] L.K. Pinpin, R.S. Johansson, C. Laschi, and P. Dario. Gaze interface: Utilizing human predictive movements for controlling a hbs. In *Proc. of the IEEE Conference on Biomedical Robotics and Biomechatronics (BIOROB)*, pages 158–162, 2008.
- [23] K. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33.
- [24] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, U.C Berkley, Computer Science Division, 2002.
- [25] N.T. Nguyen, H.H. Bui, S. Venkatesh, and G. West. Recognizing and monitoring high-level behaviors in complex social environments. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 620–625, 2003.
- [26] A.E. Nicholson and J.M. Brady. Dynamic belief networks for discrete monitoring. *IEEE Transactions on Systems, Man and Cybernetics*, 24:1593–1610, 2002.
- [27] K. Nummiaro. An adaptive color-based particle filter. *Image and Vision Computing*, 21:99–110, 2003.
- [28] A. Oliva, A. Torralba, M.S. Castellano, and J.M. Henderson. Top-down control of visual attention in object detection. In *Proc. of International Conference on Image Processing (ICIP)*, pages 253–256, 1999.
- [29] C.A. Rothkopf, D. Ballard, and M. Hayhoe. Task and context determine where you look. *Journal of Vision*, 7:1–20, 2007.
- [30] C. HintemÄ½ller C. Kapeller C. Guger M. Slater K. J. Blom S. Kishore, M. Gonzalez Franco. Comparison of ssvp bci and eye tracking for controlling a humanoid robot in a social environment.
- [31] A.P. Shon, D.N. Grines, C.L. Baker, M.W. Hoffman, Z. Shengli, and R.P.N. Rao. Probabilistic gaze imitation and saliency learning in a robotic head. In *Proc. of the IEEE International Conference in Robotics and Automation (ICRA)*, 2005, pages 2865–2870, 2005.
- [32] N. Sprague and D. Ballard. Eye movements for reward maximization. In *In Advances in Neural Information Processing Systems 15*. MIT Press, 2003.
- [33] Y. Sun, R. Fisher, F. Wang, and H.M. Gomes. A computer vision model for visual object based attention and eye movement. *Computer Vision Image and Understanding*, 112:126–142, 2008.
- [34] R. Yong and C. Yunquiang. Better proposal distributions: object racking using unscented particle filter. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 786–793, 2001.
- [35] C. Yu, D.H. Ballard, and S. Zhu. Attentional object spotting by integrating multimodal input. In *Proc. of the Fourth IEEE International Conference on Multimodal Interfaces*, pages 287–292, 2002.
- [36] L. Zhang, C. Yuanda, M. Zhang, and Y. Wang. Object tracking based on visual attention model on particle filter. *International Journal of Information Technology*, 11:109–118, 2005.
- [37] G. Zweig. A forward-backward algorithm for inference in bayesian networks and an empirical comparison with hmms. Master's thesis, Computer Science Department, UC Berkely, 1996.

Appendix A

The white vertical lines indicate the frame number at which the object has begun to move towards the next contact point; this ground truth was obtained by manually inspecting the video frame by frame. For ease of comparison, the Gaussian distributions nearest to the three contact points were labeled as 1, 2, 3; these correspond to contact points 0, 2, and 1 respectively (refer to Figure 1 for the locations of the contact points). These are shown as the top three rows of the distribution. The inferred hidden state is the discrete number of the observed value.

(a) Inferred posterior distribution for task 1 using an AHMM model without gaze observation. The task contact point sequence is 0 (home), 2, 0, 1, 0, 2, 0, 1, 0, 2, 0, 1, 0, 2, 0 (refer to Figure 2 for the location of the contact points). The inferred S_t sequence just after the transitions (white vertical lines) is 1, 2, 1, 3, 1, 2, 1, 3, 1, 2, 1, 3, 1, 2, 1, 3, 1.

(b) Inferred posterior distribution for task 2 using an AHMM model without gaze observation. The task contact point sequence is 0 (home), 2, 1, 0, 2, 1, 0, 2, 1, 0, 2, 1, 0 (refer to Figure 2 for the location of the contact points). The inferred S_t sequence just after the transitions (white vertical lines) is 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1.

Figure 9: Inferred posterior distributions $P(S_{0:t}|Y_{1:t})$ for Task 1 and 2 from AHMM models without gaze. The horizontal axis is the frame number. At each frame, the vertical column represents the value $P(S_t|Y_{1:t})$ with probability values for each possible discrete value indicated by color.

Figure 10: Inferred posterior distribution $P(S_{0:t}|Y_{1:t})$ for task 1 using an AHMM model without gaze observation. The task contact point sequence is 0 (home), 2, 0, 1, 0, 2, 0, 1, 0, 2, 0, 1, 0, 2, 0 (refer to Figure 2 for the location of the contact points). The inferred S_t sequence just after the transitions (white vertical lines) is 1, 2, 1, 3, 1, 2, 1, 3, 1, 2, 1, 3, 1, 2, 1, 3, 1.

Appendix B

Figure 11: Inferred posterior distributions $P(S_{0:t}|Y_{1:t})$ for Task 1 from AHMM models with gaze observation. The task contact point sequence is 0 (home), 2, 0, 1, 0, 2, 0, 1, 0, 2, 0, 1, 0, 2, 0 (refer to Figure 2 for the location of the contact points). The inferred S_t sequence just after the transitions (white vertical lines) is 1, 2, 1, 3, 1, 2, 1, 3, 1, 2, 1, 3, 1, 2, 1, 3, 1. The horizontal axis is the frame number. At each frame, the vertical column represents the value $P(S_t|Y_{1:t})$ with probability values for each possible discrete value indicated by color. The white vertical lines indicate the frame at which the object begins to move towards the next contact point.

(a) Inferred posterior distribution $P(S_{0:t}|Y_{1:t})$ for task 2 using an AHMM model with gaze observation. The task contact point sequence is 0 (home), 2, 1, 0, 2, 1, 0, 2, 1, 0, 2, 1, 0 (refer to Figure 2 for the location of the contact points). The inferred S_t sequence just after the transitions (white vertical lines) is 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1.

(b) Inferred posterior distributions $P(S_{0:t}|Y_{1:t})$ for Task 3 using an AHMM model with gaze observation. The task contact point sequence is 0 (home), 2, 1, 0, 2, 1, 0, 2, 1, 0, 2, 1, 0 (refer to Figure 2 for the location of the contact points). The inferred S_t sequence just before the transitions (white vertical lines) is 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1.

Figure 12: Inferred posterior distributions $P(S_{0:t}|Y_{1:t})$ for Task 2 and 3 from AHMM models without gaze. The horizontal axis is the frame number. At each frame, the vertical column represents the value $P(S_t|Y_{1:t})$ with probability values for each possible discrete value indicated by color. The white vertical lines indicate the frame at which the object begins to move towards the next contact point.