Effective Robot Task Learning by Focusing on Task-relevant Objects

Kyu Hwa Lee, Jinhan Lee, Andrea L. Thomaz, Aaron F. Bobick
Center for Robotics and Intelligent Machines
Georgia Institute of Technology, Atlanta, GA, USA 30332
{kyu, jinhlee, athomaz, afb}@cc.gatech.edu

***Abstract*— In a Robot Learning from Demonstration framework involving environments with many objects, one of the key problems is to decide which objects are relevant to a given task. In this paper, we analyze this problem and propose a biologically-inspired computational model that enables the robot to focus on the task-relevant objects. To filter out incompatible task models, we compute a Task Relevance Value (TRV) for each object, which shows a human demonstrator's implicit indication of the relevance to the task. By combining an intentional action representation with 'motionese' [2], our model exhibits recognition capabilities compatible with the way that humans demonstrate. We evaluate the system on demonstrations from five different human subjects, showing its ability to correctly focus on the appropriate objects in these demonstrations.**

I. INTRODUCTION

Robot Learning from Demonstration (LfD) has been widely studied over the past decade with the aim of providing an efficient means of teaching tasks to robots [5]. Instead of explicitly programming the required sequence of actions, it is intended that human users teach robots in a more natural way. Achieving this capability is, of course, quite challenging (as discussed in [12], [13]), and it has often been characterized as solving the following questions: *who* to imitate, *when* to imitate, *how* to imitate, *what* to imitate, and *how to judge* if an imitation was successful [9].

In this paper, we focus on the issue of *what* to imitate. As stated in [14], it differs from *how* to imitate (e.g. [10], [11]) because the robot does not intend to copy the exact trajectories of actions, but to deduce the intention of the demonstrator (e.g. [1], [17], [18]). It is known that humans tend to interpret actions based on goals rather than motion trajectories [15], [16] and the aim of Learning from Demonstration is that a human instructor should be able to teach a robot in a similar manner as she would teach a human.

In particular, we analyze the problem of finding task-relevant entities, i.e. which entity matters to the current task. *Entity* is the generalization of the conventional meaning of *object* which usually refers to a physical element to be manipulated by either the human user or the robot to achieve the given task. Thus, entity may include not only inanimate objects but also human beings, since there are tasks where the relationship between the robot and human user needs to be defined, such as *handing* over a coffee cup *to the person*.

Our work is motivated by the work done in [3], where Nagai and Rohlfing analyze *motionese*, a concept recently introduced in the field of developmental learning [2]. Motionese is the phenomenon where a teacher or parent modifies his/her behavior when demonstrating a new task or skill to a child. They exaggerate and repeat movements in order to help infants understand the key elements of actions and tasks. As this work suggests, motion is a natural and effective cue for humans to suggest which object is more important than the other.

In their work they demonstrated the kind of information that motionese communicates to a learner. In our work here, we utilize this in the domain of task learning. Our system combines this notion of motion saliency with intentional action understanding framework to help the robot focus on task-relevant objects during a demonstration. We implement an intentional action understanding mechanism using the HAMMER architecture proposed by Demiris and Khadhouri [4].

In this work, we emphasize the importance of focusing on task-relevant objects and propose a computational model that is capable of recognizing human task demonstrations in a situation where action perception inherently involves ambiguity. This paper makes the following key contributions.

1) We present a new algorithm that can quantitatively measure the relevance of entities to a task from a bottom-up based saliency map and can maintain and update those relevances over time. Defining saliency at entity level instead of at pixel level provides a learning system with more natural perception interface to a working environment.

2) We present a new computational model that integrates and augments saliency of entities with intention assertions of HAMMER to permit the description of a task as a sequence of predefined primitive operations where each operation only refers to the relevant entities. Integration of the task relevance value of entities into HAMMER architecture enables a learning system to filter out incompatible actions and thus results in reducing the level of ambiguity inherent in cluttered working space environments.

3) Last but not least, we demonstrate in our evaluation with human subjects that this computational model represents how people naturally demonstrate two object-oriented tasks to a robot.

II. SYSTEM ARCHITECTURE

A. Target hardware

We designed the system to eventually run on our upper-torso humanoid robot, Simon. Simon has 42 degrees of

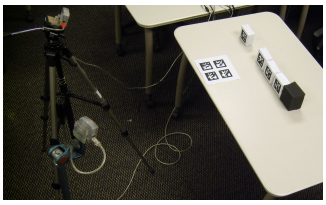


Fig. 1. System overview. Two cameras are pointing down to a desk and observing a human instructor demonstrating a task. Physical synthetic markers are applied on objects and placed on a desk to estimate pose of cameras and objects.

freedom, including two arms, a torso and a socially expressive head. Simon is able to manipulate simple objects using its hands. In this work we are developing the recognition capabilities that could form the basic functionality for Simon's task learning abilities. Although these primitive actions described in this paper are far from sufficient to learn a wide variety of tasks, they are generic enough to illustrate our approach to saliency based object selection. In this work, we use two Fire-i IEEE1394 cameras for object tracking in the workspace. As a framework, we have modified ARToolKitPlus[8] to control the camera directly using OpenCV library[19]. Although this simplifies some of the visual processing, the architecture incorporates all the necessary components for testing on real robots. Additionally, since Simon is not a mobile robot, it is reasonable to use static environmental cameras as perceptual inputs.

B. Object Detection and Localization

We detect and localize objects by estimating 6-DOF pose (3D in location and 3D in orientation) of synthetic physical markers — affixed to the faces of objects — using ARToolKitPlus [8]. We define the global reference frame using four markers on the table as shown in the left-bottom part of Fig. 1. The center of those markers becomes the origin of the reference frame and with the help of ARToolKitPlus we register the 6-DOF poses of cameras. To estimate the pose, especially the depth range, of an object, multiple cameras are used. The depth range of an object estimated with one camera is not accurate because ARToolKitPlus infers the depth range of an object by using apparent size. As the object gets farther away from the camera, the estimated scale becomes less accurate. Thus, we use multiple cameras to better estimate the pose of markers by fusing estimates obtained from each camera. We compute two line equations in a 3D space where each line joins the point of the camera itself and the estimated location of a marker from that camera. We find a point such that the sum of the distances between the point and two lines are minimum. We conduct all experiments in this paper with two cameras.

III. FINDING THE OPTIMAL SET OF ENTITIES FOR TASK LEARNING

Knowing which entities are involved in the task is critical to the efficiency of task learning and the complexity of the learned result. If the robot focuses on all the known objects in the workspace where only one or two objects are actually engaged in the task, it would result in generating a too specific and brittle task description.

In this section, we analyze three methods that influence the selection process of task relevant entities: 1) Measurement of task relevance value (TRV) of an entity, 2) Understanding a demonstrator's intention, and 3) Explicitly indicating an entity as important by means of shaking or waving at the beginning of the task. 1) and 3) are similar to the notion of motionese which is used to draw an infant's attention and emphasize particular objects at particular times in the task [3]. The output of our system is a sequence of observed actions that represents a task demonstration.

Before describing these methods, we define *Entity* as any system-recognizable object which can be either animate or inanimate. In the domain of task learning, an entity is often a physical element which is used to represent a task along with actions. In our experiments, an entity is a block on which a synthetic marker is applied and has two attributes, 3-D pose and label. Values of these attributes are obtained and updated by the marker-based detector explained in Section $II - B$.

A. Measurement of Task Relevance Value(TRV)

The main purpose of using the TRV is to give priority to the objects that are likely to be relevant to the task. An object's TRV is increased when the demonstrator starts manipulating the object and gets decreased as the object is no longer used by the demonstrator. This mechanism is particularly useful when there are many objects and only a small portion of them are used during the performance since the learner does not have to pay full attention to all of the objects in the scene. This is a scenario that will be encountered often in real-world cluttered workspaces.

To provide this functionality, we have implemented the visual attention system suggested in [7]. Instead of using color, intensity, and orientation channels, we are only using the motion channel which is relevant to our experiment. From the vector image of optical flow, the saliency map is constructed and the salient regions are located using the computation method suggested in [7].

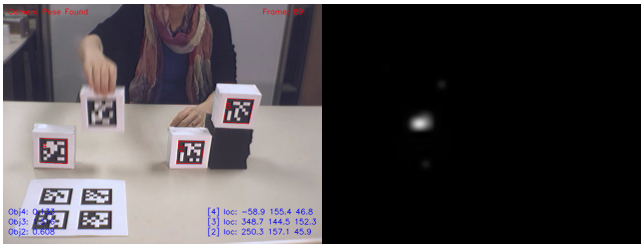


Fig. 2. Example of motion saliency. The only block that a demonstrator is moving becomes salient.

For each time frame, the saliency values in the region of the entities on image coordinates are summed and added to the corresponding object's TRV as in (1).

It is important to note that the TRVs of all entities in the scene could be considered as a *cognitive* attention map (vs.

visual attention map,) where the saliency is assigned to each *entity* instead of each *pixel* on an image. The TRV of an entity is calculated as follows:

$$TRV_t = TRV_{t-1} + \frac{\alpha * I_m}{S} - \gamma \quad (1)$$

where I_m is the sum of the normalized motion intensity values of the region in the visual attention map in which the object is occupying, S is the area of the object region, and t is the current time step. α and γ are constants that determine the rate of increase and decrease of the stimulation, respectively. In this experiment, these values were determined empirically where $\alpha = 10$ and $\gamma = 0.02$.

If TRV of an entity is under a threshold, the value which essentially comes from the noise generated in real environments, that entity is not considered in the task demo representation. Otherwise, the entity is considered important (task-relevant) and recorded into the task representation.

Clearly the parameters α and γ must reflect the timing or pace at which a task is demonstrated. If the decay rate γ is too large, all known objects in the scene will be treated with the same importance. Alternatively, if it is too small, any object that was moved once will be remembered and their relations with other objects will be recorded throughout the demonstration.

B. Understanding a Demonstrator's Intention

Understanding a human's intentional behavior could provide additional 'top-down' information about which objects the demonstrator may be focusing on. In this section, we give a brief description of the actions we used and how we generate a prediction of those acts.

1) Primitive actions: We have defined a set of object-oriented actions that could eventually be performed by our robot. They include `moveUp`, `moveLeft`, `moveDown`, `moveRight`, `placeLeftOf`, and `placeRightOf`. The notion of `placeRightOf` and `PlaceLeftOf` does not strictly have a meaning of *put on the ground*. It can simply mean *being proximate to* either left or right side of another object.

2) Action recognition: In this work, a Hierarchical Attentive Multiple Models for Execution and Recognition (HAMMER) model is used to estimate the demonstrator's intention [4]. Motivated by theories of cognitive function that focus on mental simulation [6], this architecture is composed of basic building blocks involving a pair of inverse and forward models that are used to either perceive or execute an action as shown in Fig. 3. $B_n(n = 1, 2, ..., N)$ is one of the N inverse models, i.e. primitive actions, and F_n is one of the N forward models, i.e. predictors. M_n, P_n, E_n are motor signal, prediction signal, and error signal, respectively.

The role of an inverse model, which could be thought of as a primitive action in our case, is to generate appropriate motor commands to the system to achieve the target goal based on the observation of the current state. The inverse models could be parametrized with the object type rather than keeping a separate model for each object. In contrast,

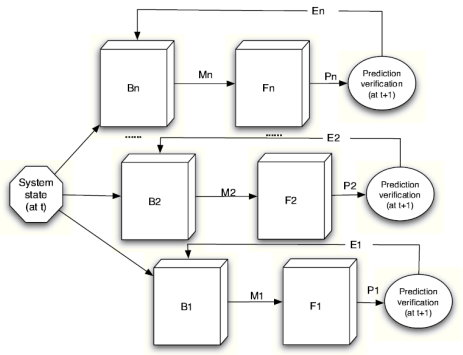


Fig. 3. Basic HAMMER architecture [4]

the role of a forward model is to output the predicted next state of the system by considering both the current state from the system and a control command. A set of inverse-forward models were hand-coded as done in [4] to output a prediction of the next state of the system.

Once the prediction is made, it is compared with the actual state of the demonstrator in the next time step and an error signal is generated. This error signal changes the confidence value of an inverse model based on (2). Hence, taking the perceived visual information as input, HAMMER hypothesizes the demonstrator's intent and computes measurable predictions of the next possible state (i.e., the robot forms expectations about how the demonstrator will move next.) It is then verified on the next time step to see whether the hypothesis was correct. Depending on the result, the confidence value *conf* of every inverse model is updated, which is equivalent to the likelihood value of an action. For further description on this architecture, please see [4].

Due to the high computational complexity that might occur from updating all of these inverse models, only the inverse models which consider the objects that are being manipulated are updated. This is important since the number of inverse models increases as the number of known objects rises.

The confidence value of the k-th inverse model could be computed as follows:

$$conf_k(t) = \begin{cases} conf_k(t-1) + 1 + N_r & \text{if prediction is correct} \\ conf_k(t-1) - 1 - N_r & \text{otherwise} \end{cases} \quad (2)$$

where t is the current time step and N_r is the number of times the inverse model has been rewarded in the past. Similar to TRV, due to the natural noise, confidence value under a threshold is set to zero, where ε = 10 in our case.

3) Task generation: The inverse model with the highest confidence value is recorded when it takes a clear lead, i.e. the maximum confidence value is larger than 2 times that of the average value μ. To quantize this value, we define $conf_{ratio}$ as follows:

$$conf_{ratio} = conf_{max}/(2 * \mu) \quad (3)$$

An action sequence is added to the task demo representation when $conf_{ratio} > 1.0$; otherwise, no action sequence is recorded.

Fig. 4. Two types of tasks to be learned by the robot. Task 1 tests effects of saliency of objects on task learning and Task 2 registration of objects as well as saliency of objects respectively.

C. Explicitly expressing an important entity

In an explicit teaching interaction, there are times that the human instructor wants to give attention to some entities explicitly. In other work, it has been shown that parents engage in this type of behavior when demonstrating to infants [3]. Our system has the ability to take the advantage of this social cue. By shaking an object, the demonstrator can force the robot to be aware of it, which we call *registration*. Once the object is registered, the object is always considered task-relevant (i.e. salient) regardless of TRV, and throughout the task demonstration it is included in the task representation. Registration could also be used to clarify an ambiguous situation, which will be discussed in the following section.

IV. EXPERIMENTS

Our system is designed to observe a person manipulating objects in the workspace, and determine the sequence of primitive actions they were meaning to demonstrate. There are several objects in the workspace, thus the particular object relations that the person means to demonstrate are ambiguous. We demonstrate that our saliency mechanism allows the system to focus attention on task relevant objects in order to infer the appropriate actions.

Since our end goal is for the robot to learn from everyday people, the appropriate evaluation of our system is its performance in recognizing task demonstrations from a number of different users (who are not system designers). This measures the extent to which our system is generic to the variety of ways that people demonstrate object oriented tasks.

A. Experimental Setup

In our experiment we test two tasks as shown in Fig. 4. Each task is performed five times each, by five volunteers, resulting in 25 demonstrations per task. Each person was given two task description cards as instructions for what to demonstrate. They demonstrated each task to the robot by repeating the same task five times. They were not instructed about how to perform the task (e.g., how fast to move the objects, etc.).

In Task 1, the demonstrator was instructed as follows:

Pick block 1 and move upward. Move it left and place it on the right side of block 4. Next, pick block 4 and move upward. Move it right and place on the right side of block 1.

In Task 2, the demonstrator was instructed as follows:

First, pick block 4 up, shake it for a short period, and place it down as if you were to indicate that this block is important. Then, move block 3 left and place it down near the gray wall. Pick block 1 and move upward, move left, and place it on the left side of block 4, which you shook last time. Pick block 1 again and move upward, move right, and place it on the right side of block 4.

B. Experimental Results

Our evaluation considers how often the system correctly classified the sequence of primitive actions for the two tasks. The results are shown in Table I. For the first task, the system correctly segmented and classified action sequences in the right order on 22 out of 25 demonstrations.

The Fig. $5(a)$ shows a case when the system successfully encoded the demonstrator's action sequence using only blocks 1 and 4.

In the last part of the Task 1, although the observed action may be encoded as either `placeRightOf(4,1)` or `placeLeftOf(4,2)`, `placeRightOf(4,1)` was selected because the user recently moved block 1. Also, even though block 1 passed the left side of block 3, which can be interpreted as `placeLeftOf(1,3)`, it was not recorded because the system identified the demonstrator's intention as `pickUp(1)`. Hence, blocks 2 and 3 are not included in the task demo represenation.

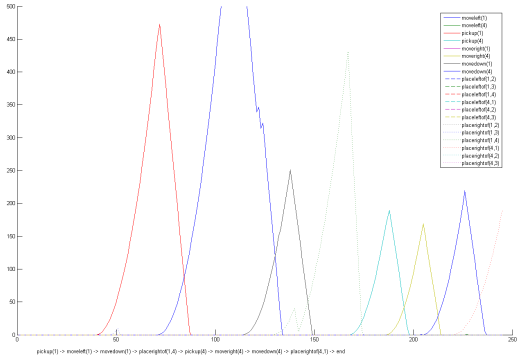
However, Fig. $5(c)$ shows a sequence where the system added `placeRightOf(1,3)` which can be seen at around frame 50, due to the ambiguity of the demonstrator's intention. In this case, block 3, which is not part of the task, was recorded in the task demo representation. The demonstrator slowly moved block 1 up such that it stayed right next to block 3 for a significant amount of time. This explains why `placeRightOf(1,3)` was added to the task demo representation.

For the second task, the system correctly classified 16 out of 25 demonstrations as shown in Table I. In the second task, demonstrators intentionally registered block 4 by shaking it at the beginning of the task. The example in $5(b)$ shows a case that when the instructor moved block 1 down and then back to the right side of block 4, the system correctly selected `placeLeftOf(1,4)` with high confidence. The system fails sometimes due to the speed of demonstration. In the example shown in Fig. $5(d)$, the system is not certain whether the last intended action was `placeLeftOf(1,3)` or `placeRightOf(1,4)`, where the correct action is `placeRightOf(1,4)`. The TRV of block 3, from the initial movement, was still high enough to consider `placeLeftOf(1,3)` when block 1 was placed between blocks 4 and 3. In this case, both `placeLeftOf(1,3)` and `placeRightOf(1,4)` show high confidence values, resulting in an ambiguous situation.

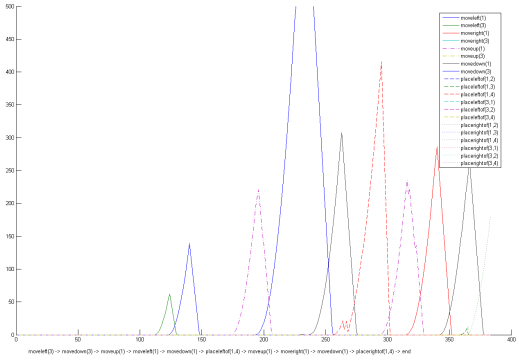
V. DISCUSSION

In this paper, we proposed a computational model that records task demonstrations in a human-like manner and reduces possible ambiguity in action perception, which can easily happen in real-world cluttered workspaces. By incorporating the mechanisms of entity saliency evaluation, intention understanding, and registration, our system is able to make reasonable estimates of the appropriate entities to include in a task demonstration example.

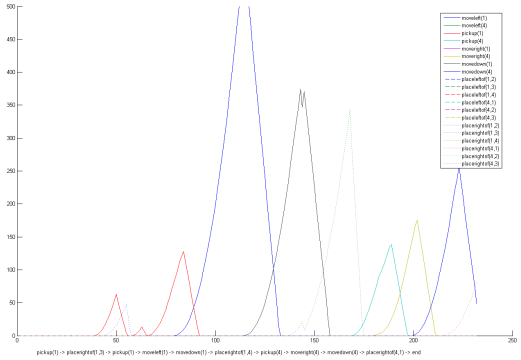
Our notion of Task Relevance Value (TRV) of entities represents the implicit indication that a demonstrator makes



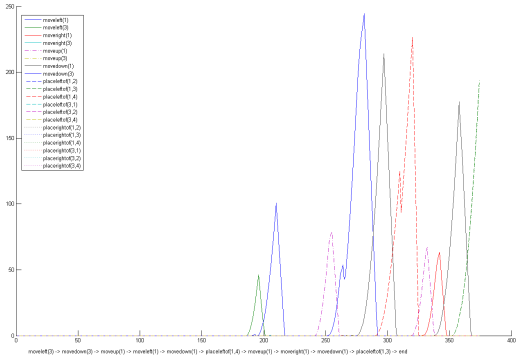
(a) Intention graph of Task 1 when successfully learned. The color shows which object the human demonstrator is currently manipulating.



(b) Intention graph of Task 2 when successfully learned. The color shows which object the human demonstrator is currently manipulating.



(c) A case where Task 1 was learned with error. At around frame 50, placeLeftOf(1,3) is recorded. It is shown in green color.



(d) A case where Task 2 was learned with error. Confidence values of both placeLeftOf(1,3) and placeRightOf(1,4) are high, which are shown in green color. The first 100 frames are not shown due to the limited space.

Fig. 5. Change of the confidence values as the demonstration unfolds. In these graphs, each peak value could be thought of as a recognized action that is recorded into the task demo representation. Only inverse models that are related to the objects manipulated by the demonstrator are considered. Correct sequence of Task1: `moveUp(1)-moveLeft(1)-moveDown(1)-placeRightOf(1,4)-moveUp(4)-moveRight(4)-moveDown(4)-placeRightOf(4,1)`. Task2: `moveLeft(3)-moveDown(3)-moveUp(1)-moveLeft(1)-moveDown(1)-placeLeftOf(1,4)-moveUp(1)-moveRight(1)-moveDown(1)-placeRightOf(1,4)`

TABLE I

POSSIBLE CASES AND RESULTS

Task 1		
Error type	Frequency	Comments
Additional actions	3	The instructor hesitated while moving the block
Confusing actions	0	Did not occur
Task 2		
Error type	Frequency	Comments
Additional actions	1	The instructor aligned blocks after the placement
Confusing actions	8	TRV remained high enough to give confusion

about the relevance of objects during their task execution. Paying attention to this important cue in human behavior allows the robot to filter out incompatible inverse models, and thus reduce the level of ambiguity inherent in cluttered workspace environments.

Our experiment shows that our system is often successful in focusing on task-relevant objects and recognizing the intended actions of a human's demonstration. During the two task examples used in this experiment, the majority of demonstrations were recognized correctly by filtering out irrelevant action sequences that are not supposed to be demonstrator's intentions. In addition, the system errs on the side of caution which in turn is less detrimental to the overall performance of learning by demonstration.

One limitation of our system is that it is difficult to find the optimal increase and decrease rates while calculating TRV. Most of the errors occurred in our experiments were due to the variability of the demonstration speed, in addition to his imprecise actions. Also, we need a predefined set of inverse models that might be tedious to write. However, by defining the minimum set of commonly used primitive operations, it would be possible to represent many higher level tasks with the combination of these operators.

It is worth noting that multiple demonstrations are necessary, allowing the robot to adjust missing or corrupted parts of the task representation over time. If communicative skills

are added, a common feature in active learning, the robot might be able to effectively correct the misidentified action segments.

VI. CONCLUSION

In this paper we have presented a computational system for robot learning by demonstration. Our focus is on the problem of how a robot determines *what* to imitate, in particular how it can determine which objects in the environment are relevant to the task demonstration. We have shown that a system that combines the use of forward and inverse models for action representation and social 'motionese' cues can efficiently record examples of task demonstrations from a human partner in ways that coincide with the task they intended to teach. In the future, we would like to incorporate the notion of context using a stochastic grammar to better identify not only irrelevant objects but also irrelevant events.

REFERENCES

[1] Yiannis Demiris and Gillian Hayes, "Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model", *in Imitation in Animals and Artifacts*, eds., K. Dautenhahn and C. L. Nehaniv, 321-361, MIT Press, 2002.

[2] Rebecca J. Brand, Dare A. Baldwin, and Leslie A. Ashburn, "Evidence for 'motionese': modifications in mothers' infant-directed action", *Developmental Science*, 5(1), 72-83, 2002.

[3] Yukie Nagai and Katharina J. Rohlfing. "Can Motionese Tell Infants and Robots 'What to Imitate'?", *In Proceedings of the 4th International Symposium on Imitation in Animals and Artifacts (in AISB'07)*, pp. 299-306, April 2007.

[4] Demiris, Y., & Khadhouri, B. "Hierarchical attentive multiple models for execution and recognition of actions", *Robotics and Autonomous Systems*, (2006), 54, 361-369.

[5] B.D. Argall, et al., "A survey of robot learning from demonstration", *Robotics and Autonomous Systems*, (2009), to be published.

[6] G. Hesslow, "Conscious thought as simulation of behaviour and perception", *Trends in Cognitive Sciences 6 (6)* (2002) 242-247.

[7] Itti, L., Koch, C., & Niebur, E., "A model of saliency-based visual-attention for rapid scene analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 1254-1259, 1998.

[8] D. Wagner, & D. Schmalstieg, "ARToolKitPlus for Pose Tracking on Mobile Devices", *Proceedings of 12th Computer Vision Winter Workshop*, February 2007.

[9] K. Dautenhahn, C.L. Nehaniv, "The agent-based perspective on imitation", in: *K. Dautenhahn, C.L. Nehaniv (Eds.), Imitation in Animals and Artifacts*, The MIT Press, Cambridge, MA, 2002, pp. 1-40.

[10] Aude Billard, "Learning motor skills by imitation: A biologically inspired robotic model", *Cybernetics and Systems: An International Jounal*, 32, 155-193, (2001).

[11] Aude G. Billard, Sylvain Calinon, and Florent Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks", *Robotics and Autonomous Systems*, 54(5), 370-384, (2006).

[12] Cynthia Breazeal and Brian Scassellati, "Robots that imitate humans", *Trends in Cognitive Sciences*, 6(11), 481-487, (2002)

[13] Cynthia Breazeal and Brian Scassellati, "Challenges in building robots that imitate people", *Imitation in Animals and Artifacts*, eds., K. Dautenhahn and C. L. Nehaniv, 363-389, MIT Press, (2002).

[14] Jansen B, Belpaeme T, "A computational model of intention reading in imitation.", *Robo. Auton. Syst.* 54(5):394-402, 2006

[15] D.A. Baldwin and J.A. Baird, "Discerning intentions in dynamic human action.", *Trends in Cognitive Sciences*, 5(4):171-178, 2001.

[16] A. L. Woodward, J. A. Sommerville, and J. J. Guajardo, "How infants make sense of intentional actions", B. Malle, L. Moses, and D. Baldwin, editors, Intentions and Intentionality: Foundations of Social Cognition, chapter 7, pages 149-169. *MIT Press*, Cambridge, MA, 2001.

[17] A. Lockerd and C. Breazeal, "Tutelage and socially guided robot learning", *in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2004.

[18] Sylvain Calinon, Florent Guenter, and Aude Billard, "Goal-directed imitation in a humanoid robot", *in Proceedings of the International Conference on Robotics and Automation(ICRA)*, (2005).

[19] G. Bradsky: "The opencv library". In *Dr. Dobb's Journal of Software Tools*(2000), pp. 122.125.