Visuo-Cognitive Perspective Taking for Action Recognition

Matthew Johnson and **Yiannis Demiris** [1]

Abstract. Many excellent architectures exist that allow imitation of actions involving observable goals. In this paper, we develop a Simulation Theory-based architecture that uses continuous visual perspective taking to maintain a persistent model of the demonstrator's knowledge of object locations in dynamic environments; this allows an observer robot to attribute potential actions in the presence of goal occlusions, and predict the unfolding of actions through prediction of visual feedback to the demonstrator. The architecture is tested in robotic experiments, and results show that the approach also allows an observer robot to solve Theory-of-Mind tasks from the 'False Belief' paradigm.

1 Introduction

When we see another person performing an action, we are usually able to understand the purpose and intention underlying the action, and can reproduce the action for ourselves. The HAMMER architecture [5, 16] can be used to equip a robot with this common human ability. The HAMMER architecture achieves the mapping between observed and self-generated action by directly involving the observer robot's motor system in the action recognition process; during observation of the demonstrator's actions, all the observer's inverse models (akin to motor programs) are executed in parallel in simulation using forward models. The simulated actions generated by the inverse models are compared to the observed action, and the one that matches best is selected as being the observed action. The internal action simulation, combined with the comparison to the observed action, achieves the mapping between observed action and self-generated action that is required for imitation [4].

By using the motor system to achieve action recognition, the HAMMER architecture is taking a Simulation Theory approach to solving the imitation problem. In the 'Theory of Mind' paradigm, the Simulation Theory is used to attribute mental states to other people by using one's own cognitive decision-making mechanism as a manipulable model of other's minds, taken off-line and placed into the context of their situation [13, 9, 8]. For this to work, the state of the 'target' agent is used instead of one's own state, but transformed into an egocentric format that our first-person decision-making and behaviour-generation mechanisms will accept.

Similarly, in order to provide meaningful data for comparison, the simulated actions used by the HAMMER architecture during recognition must be generated as though from the point of view of the demonstrator. Since the HAMMER architecture uses a Simulation Theory approach, the observer's inverse models require first-person data in order to generate actions, and so spatial and visual perspective taking are used to achieve the egocentric 'shift' from the observer to the demonstrator. The data required for the inverse models to operate is therefore derived from consideration of the demonstrator's physiospatial and sensory circumstances, and not the observer's, using perspective taking [11].

However, it is not only instantaneous sensory information that informs goal selection and action planning. It is through keeping in memory details of objects that are seen that *cognitive maps* are built up, which are critical to action generation. In this paper, we present a Simulation Theory approach to perspective taking that allows an observer robot to use its visual perceptual mechanisms in simulation to determine what the demonstrator is seeing; by performing this process continually, the observer's first-person cognitive map generation routines can be used to build up and maintain a representation of the demonstrator's own cognitive map. Taking into account the demonstrator's knowledge of the world in this manner allows more accurate state and goal information to be fed to the HAMMER architecture.

2 Background

In common and also academic use, the term 'perspective taking' has many meanings in many different situations. There are such definitions as:

- "People's ability to experience and describe the presentation of an object or display from different vantage points" [1]
- "Imagining oneself in another's shoes" [7]
- "Understand[ing] how others perceive space and the relative positions of objects around them- [...] the ability to see things from another person's point of view" [15]
- "Consider[ing] the needs and wants of the opponent" [6]

In this paper we focus on equipping robots with perceptual and cognitive perspective-taking abilities, through a Simulation Theory approach, in order to improve the quality of the state information fed to the HAMMER architecture. In this architecture, a cognitive map is defined as being a *representation in memory of the location of observed objects*. This memory is updated continually from observation of the environment, and is available to the action generation system for action planning. The cognitive map is used also as a manipulable spatial model of the environment to facilitate perspective taking; to enable visual perceptual perspective taking, the objects in the cognitive map are linked with visuo-spatial representations that are used to re-create the visual image seen by the demonstrator.

[1] BioART, ISN Group, Department of Electrical and Electronic Engineering, Imperial College London. Email: {matthew.johnson, y.demiris}@imperial.ac.uk

in Proceedings of the AISB-2007, pp. 262-269, (2007)

2.1 HAMMER

The HAMMER (Hierarchical Attentive Multiple Models for Execution and Recognition) architecture is a Simulation-Theoretical architure for action recognition and imitation, based on the hierarchical coupling of internal models to produce simulation loops. HAMMER achieves first-person action generation using coupled *inverse* and *forward* models, and uses the same arrangement to achieve imitation, but fed from a perceptual perspective-taking process involving internal inverse and forward *vision models*. The perceptual perspective taking process has been shown to improve the performance of action recognition in situations where the observer must take into account visual occlusions and visual cues provided to the target [11].

2.2 Internal Inverse and Forward Models

One of the core components of the HAMMER architecture is the *inverse model* for motor control. Inverse models represent functionally specialised units for generating actions to achieve certain goals. The generic inverse model takes as input the *current state* of a system, a *goal state* that is the system's desired state, and produces as output the action required to move the system from its current state to the goal state [12, 18]. In the control theory literature, the inverse model is known as a *controller* and its outputs are control signals; when applied to robotics, the current state is the state of the robot and its environment, and the outputs are motor commands. In that context, inverse models are known as *behaviours*.

Inverse models have several *internal states*, that are used in action execution and recognition [10]. One of these states is the *applicability* of the inverse model. When presented with a goal, the inverse model will calculate its level of applicability through simulation with its coupled forward model. The applicability is a measure of how useful the inverse model is for achieving the goal. A low applicability level means that the inverse model cannot achieve the goal from its current state, for example, the "Place object on shelf" inverse model when the shelf is too high to reach. The applicability level is explained in more detail in Section 3.3.

Forward models of causal dynamics are used in predictive control systems. The classic forward model takes as input a system state and the dynamics currently acting on the system, and produces as output the *predicted next state of the system*. In the HAMMER architecture, multiple forward models are *coupled* to inverse models to create a simulation process. This approach is similar to that used in other internal model-based systems for motor control [21, 20]. When coupled to an inverse model, a forward model receives the action output from the inverse model; the forward model then generates a prediction of the state that would result, if the action was to be performed.

2.3 Inverse and Forward Vision Models

In [11], the capacity for *visual perceptual perspective taking* was introduced to the HAMMER architecture. In keeping with the simulation theoretical approach, this was achieved through a biologically inspired *simulation of visual perception*. In the same way as action recognition and imitation is achieved in the HAMMER architecture through coupled inverse and forward models as used in control, visual perception and perspective taking is performed here through coupled inverse and forward models of the *visual* process. The *inverse vision model* is defined as taking two inputs, the first being a camera image, and the second being the visual parameters with which to process that image. The output from the model is the state

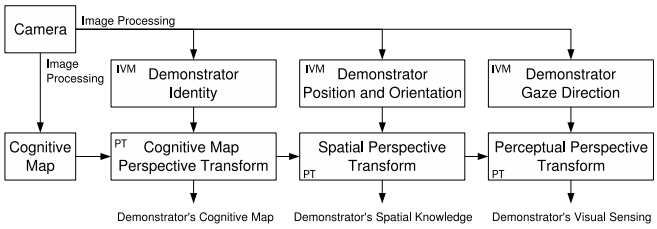


Figure 1. The perspective-taking process. Image information from the camera, and the robot's own knowledge held in the cognitive map, are fed into a cascade of perspective transform 'filters'. The outputs at each stage are used as the 'pretend states' fed into the HAMMER architecture. 'PT' indicates a perspective transform stage, and 'IVM' indicates an inverse vision model for performing image processing.

output from processing. A *forward vision model* is defined as having two inputs and one output. The forward vision model takes as input visual object properties retrieved from the cognitive map (e.g. colour, shape, etc), and their desired state (e.g. positions and orientations taken from the cognitive map), and produces as output the visual image that results from reconstructing these inputs. Inverse and forward vision models are described in detail in [11].

The coupling of inverse and forward vision models results in simulation of perception, and gives HAMMER the ability to consider what the demonstrator *sees*, as well as its position. This enables the observer robot to take into account visual occlusions effecting the demonstrator, and through *continual* usage, the observer can keep track of what objects the demonstrator has seen in the past and potentially stored in its cognitive map. Because the demonstrator sees different things to the observer due to its differing viewpoint, it becomes necessary for the observer robot to maintain a representation of the demonstrator's cognitive map in order to predict and recognise actions. In keeping with the simulation theory approach, this may be achieved by recruiting the observer's own cognitive map creation and updating processes, but fed with information derived from visual perceptual perspective taking instead of first-person visual information. Figure 2 shows the perception simulation process.

3 Implementation

The perspective taking architecture shown in Figures 1 and 2 was implemented in C++ for experiments involving an observing observer robot and a demonstrator robot. The target robots were ActivMedia Peoplebots, equipped with grippers and firewire cameras. A version of HAMMER was implemented and linked to the perspective taking architecture.

3.1 Inverse and Forward Vision Models

Inverse vision models were implemented using the ARToolkit Plus, an extension of the ARToolkit [2]. When presented with an image containing two-dimensional square markers (fiducials) of known size, the ARToolkit can calculate the position and orientation of the markers in world co-ordinates. A set of three objects was therefore produced with fiducials attached, and in order to extract the demonstrator robot's position and orientation at any point in time, a cubic AR 'hat' was made, with a fiducial on each vertical face. This ensured that no matter which direction the demonstrator robot was fac-

ing, the observer robot would be able to determine its location and orientation.

To construct visual scenes from the transformed cognitive map, the forward vision models used the OpenGL graphics library (www.opengl.org). To ensure that the same inverse vision models as used for first-person visual processing would work with the reconstructed image for the demonstrator robot, the fiducials used by the ARToolkit were added in as OpenGL textures and linked to the object entries in the cognitive map.

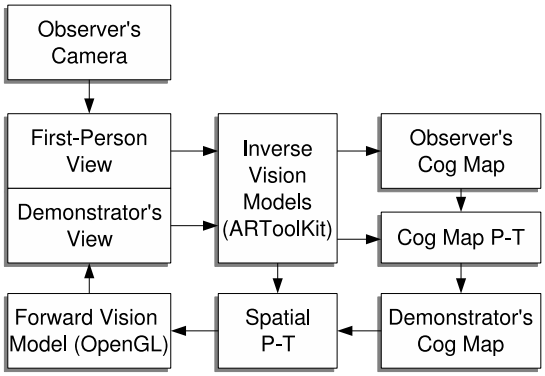


Figure 2. The perception simulation loop. The observer's first-person view of the scene is used to build up the observer's cognitive map of the scene. The cognitive map is filtered through the cognitive map perspective transform that 'filters' the observer's cognitive map to make it like the demonstrator's. This is then used as a basis for the perceptual perspective transform, that begins with a spatial geometric transform to 're-centre' on the demonstrator, and then fed through the forward vision model to re-create what the demonstrator is seeing. The observer can then use its inverse vision models on the image to update its representation of the demonstrator's cognitive map, in the same way as it would update its own.

3.2 Cognitive Map Definition

The cognitive map was defined as being a list of objects, held in memory. It was assumed that the robots already knew what each object was and could identify them through the inverse vision models (i.e. the inverse vision models were programmed to recognise the objects, through use of the fiducials, and extract relevant information). When visual information for the objects was available from the inverse vision models, the cognitive map entries for those objects were updated with world position and orientation values. Linked with this information was a three-dimensional model of each object, and visual information (e.g. colour and texture) that would be used by the observer's forward vision model to re-create the image of the scene from the point of view of the demonstrator. As can be seen from Figure 1, the perspective-taking process is then comprised of the following steps:

1. The demonstrator is identified and the correct cognitive map perspective transform, comprising the differences from the observer's own cognitive map, is applied;
2. A spatial perspective transform is applied to the resulting cognitive map, to re-centre it upon the demonstrator;
3. The forward vision model takes in the re-centred spatial data, and accesses the visual information linked to the objects in the cognitive map, to re-construct the image that the demonstrator is seeing. This image is then processed by the observer's inverse vision models, to update the demonstrator's cognitive map transform, and to provide state information to HAMMER.

3.3 Inverse and Forward Models

Inverse models for the HAMMER architecture were implemented as PID controllers, generating robot velocities and delta-angle headings in order to minimise the distance between the robot grippers and a goal object. The state information required for the inverse models was taken from either the observer's own cognitive map, or its representation of the demonstrator's cognitive map. When used for action simulation, the applicability level A_t of the inverse model was calculated for the n^{th} simulation iteration according to:

$$A_{t,n} = \begin{cases} 0 & \text{at } n = 0 \\ A_{t,n-1} + \frac{1}{S_d} \times \frac{1}{n} & \text{if } \frac{dS_d}{dt} < 0 \\ A_{t,n-1} - \frac{1}{S_d} \times \frac{1}{n} & \text{if } \frac{dS_d}{dt} \geq 0 \end{cases} \quad (1)$$

The applicability accumulation is discounted over time and is increased (rewarded) if the inverse model is making progress towards achieving its goal, and decreased (punished) if it is not. The state distance between current state S_t and the goal state vector λ is defined as:

$$S_d = \sum_{i=1}^{M} |\lambda_i - S_{t,i}| \quad (2)$$

When S_d was less than a completion threshold ϵ_1, the inverse model became complete and did not generate motor commands even when instructed to execute. In the following experiments, ϵ_1 was chosen to be 0.04.

Forward models used Euler integration to form 1-timestep predictions based on the current state, and the robot velocity and heading motor commands generated by the inverse models. As in [11], the forward models were equipped with collision models of the robot and objects in the environment, allowing the forward model to predict position and velocity states in situations when the robot ran into tables or other objects.

3.4 Perspective Taking Visual Representations — 'Ghosts'

Through coupling the perception simulation loop to the simulation of action enabled by the HAMMER architecture, it is possible to *predict the visual feedback* arising from the action. By processing this visual feedback with the inverse vision models, and updating the cognitive map representation, the action simulation can continue further into the future, and the *outcome* of actions predicted. In section 4, experiments are described in which this approach is used to create multiple *Perspective Taking Visual Representations* (PTVRs), parallel instances of the observer's own perception and action mechanisms, each driven by a different inverse model. Using perspective taking, the observer can place these 'ghosts' in the place of the demonstrator, and use them to predict what the visual feedback will be from possible actions the demonstrator may perform. This allows the observer to predict the changes to the demonstrator's knowledge of the world during the course of a possible action, and how this may effect the course of the action. Figure 3 shows the arrangement.

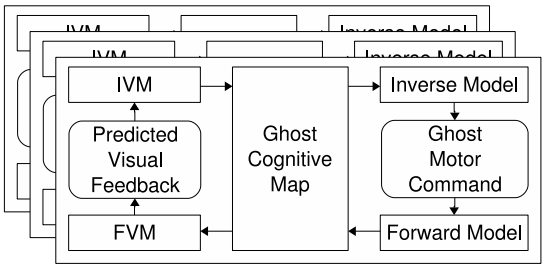


Figure 3. Perception and action simulation coupled for the generation of PTVRs ('ghosts'). Multiple ghosts can be instantiated and used in parallel, each one driven by a different inverse model. By coupling the HAMMER action simulation loop to the perception simulation loop, it is possible to predict the visual feedback to the ghost and therefore the updates to the ghost's cognitive map. A ghost can represent either the observer or the demonstrator performing a certain action.

4 Experiments

The implemented perspective-taking architecture was deployed onto the robots arranged in the scenario shown in Figure 4.

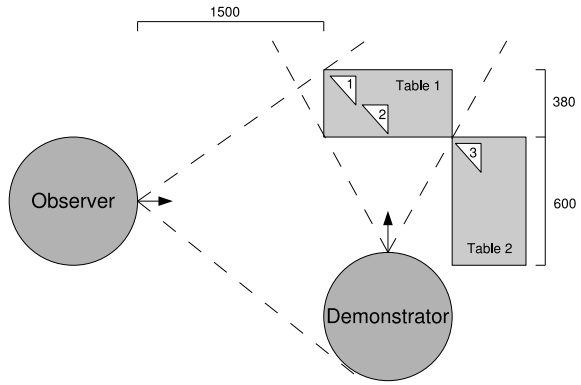


Figure 4. Plan view of the experimental setup. The observer robot faces two tables. Table 1 has objects 1 and 2, and table 2 has object 3. The demonstrator robot is positioned so that it, and the tables and the objects, can be clearly seen by the observer. The demonstrator is placed facing and close to table 1, so that initially it is able only to see objects 1 and 2. The dashed lines indicate the fields-of-view of the robots. The plan is not to scale, but measurements have been provided to indicate size and relative position. Measurements are in millimetres. The objects were 150mm across their long edge. The ARToolKit fiducials had 120mm edge, and were mounted as per Figure 5.

Three ARToolKit fiducials were attached to triangular objects, to enable both the observer and the demonstrator to identify and locate those objects. The objects were placed so that both the observer and the demonstrator could see the objects, however, initially, the observer could see all three objects whereas the demonstrator could see only the first two. The objects and the demonstrator robot were given simple 3D models to enable their reconstruction into the image produced by the forward vision model. Similarly, the implemented HAMMER architecture was provided with three 'nudge object' inverse models, one for each of the objects used during the experiments. These inverse models, when activated, produced motor commands for moving the robots from their current position to the specific object, and stopped when the robot gripper touched the object.

During the experiments it was assumed that the demonstrator's camera was kept stationary with respect to the robot's frame, pointing directly ahead of the robot.

4.1 Experimental Scenarios

The first experiment was designed to test the architecture's ability to update its cognitive map representations, in the presence of demonstrator movements, and object movements both seen and unseen by the demonstrator. There were three parts:

1. The observer takes the perspective of the demonstrator robot, and initialises its representation of the demonstrator's cognitive map;
2. Demonstrator rotates 45 degrees to its right. The observer, through continual perspective taking, updates its cognitive map representations;
3. Object 1 is moved, unseen to the demonstrator, but seen by the observer.

Experiment Two took this further, by having the observer maintain its cognitive map representations over five episodes of object and demonstrator movements, in which objects were occluded from both the demonstrator and the observer. The observer also had to use its cognitive map representations to attempt to determine what actions the demonstrator *believed* it could perform, through using perspective taking and action simulation to calculate inverse model applicabilities. The sequence was the following:

1. Demonstrator can see objects 1 and 2. Object 1 is not graspable, and the demonstrator cannot see object 3 (Figure 4);
2. Object 1 is moved close to the demonstrator, occluding object 2 from the observer;
3. Demonstrator rotates 45 degrees to its right. Object 3 becomes observable (as per Figure 7 B);
4. Object 1 is moved back to original position. The observer can see this, but the demonstrator cannot;
5. Demonstrator rotates back to original position.

In Experiment Three the observer robot had to maintain its cognitive map representations over four episodes of *simultaneous* demonstrator and object movements. The observer also had to predict the visual feedback during each potential demonstrator action using its PTVRs, in order to predict the impact of false beliefs on the performance of the actions. The sequence was the following:

1. Demonstrator can see objects 1 and 2. Object 1 is not graspable, and the demonstrator cannot see object 3 (Figure 4);
2. Demonstrator rotates 45 degrees to its right, then object 1 is moved to a graspable position (unseen by demonstrator);
3. Objects 1 and 2 are moved away (unseen by demonstrator);
4. Demonstrator rotates back to original position. Object 3 moved away (unseen by demonstrator).

5 Results

5.1 Experiment 1

Figure 6 shows the observer's view of the scene during part 1 of the experiment. The demonstrator robot and the objects are visible. Figure 6 A shows the thresholded camera image fed to the observer's inverse vision models, and Figure 6 B shows the resulting reconstruction of the visual scene, using data from the observer's cognitive

map and its forward vision model. The ARToolKit has successfully extracted the position and orientation of the objects and the demonstrator, and Table 1 shows the contents of the observer's cognitive map resulting from the processing. The X, Y, Z position and angle of objects are extracted and updated in the observer's cognitive map representations while the objects are visible.

Table 1. Cognitive map entries for centroid positions and orientations of objects when viewed by the observer robot (first-person perspective). The values shown are relative to the observer's camera position and orientation. The results correspond to the scene shown in Figure 6.

Object	**X** (m)	**Y** (m)	**Z** (m)	**Angle** (Degrees)
Demonstrator	-0.56	0.20	1.80	357.51
Object 1	0.32	-0.28	1.61	44.03
Object 2	0.20	-0.27	1.75	50.39
Object 3	-0.15	-0.24	2.21	12.13
Table 1	0.37	0.00	1.75	90.00
Table 2	-0.15	0.00	2.30	0.00

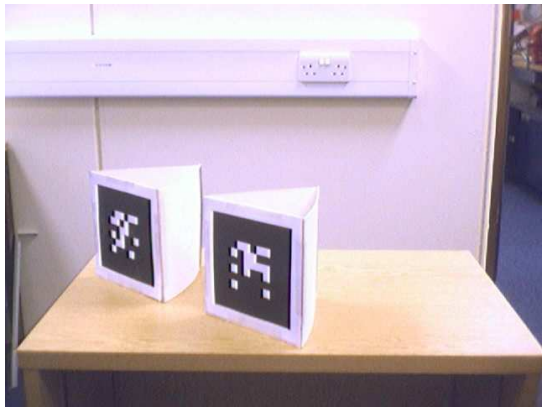


Figure 5. The demonstrator's view of the table and objects 1 and 2. Figure 7 (A) shows an internal simulation of this viewpoint by the observer.

Figure 7 shows the result of the perceptual perspective taking. Figure 7 A is what the observer determines the demonstrator to be seeing during the first part of the experiment; Figure 5 shows the demonstrator's actual camera image of this scene—the simulation of perception has clearly resulted in accurate perspective taking. Objects 1 and 2 are observed, but object 3 is outside the field-of-view on the table to the right. Figure 7 B shows the scene during part 2, after the demonstrator has rotated 45 degrees to the right. The observer robot realises through perspective taking that object 3 is now visible to the demonstrator, and objects 1 and 2 are not.

Table 2 shows the results from part 3 — moving object 1 while it can be seen by the observer, but not by the demonstrator. Through perceptual perspective taking the observer knows that the demonstrator cannot see the object being moved — and so, it updates its *own* cognitive map with the change in position, but not the demonstrator's. This leads to the discrepancy between the object 1 position values for the observer and the demonstrator, as shown in the table. The demonstrator *believes* that the object is in the place where it last saw it, whereas the observer *knows* it to be somewhere else.

Table 2. Cognitive map entries for observer and demonstrator after movement of object 1 inside the observer's field of view but outside of the demonstrator's. Through perceptual perspective taking, the observer knows that the demonstrator cannot see object 1 while it is being moved, and so the demonstrator's cognitive map is not updated with the changes in position and orientation as the object is moved.

Object	**X** (m)	**Y** (m)	**Z** (m)	**Angle** (Degrees)
Observer's Cognitive Map				
Object 1	0.12	0.06	0.55	20.96
Object 2	0.24	-0.26	1.82	55.07
Object 3	0.05	-0.23	2.17	40.86
Demonstrator's Cognitive Map				
Object 1	0.26	-0.03	0.83	64.40
Object 2	0.24	-0.26	1.82	55.07
Object 3	0.05	-0.23	2.17	40.86

5.2 Experiment 2

Building on the success of Experiment 1, the perspective taking was linked to the HAMMER architecture for action simulation experiments. Figure 8 shows the results of the applicability calculations for Experiment 2. Each episode is separated by a period of zero applicability, before the action simulations begin and then reset five seconds later. Figure 8 A shows the applicability levels when the observer is drawing on its representation of the demonstrator's cognitive map to generate state information for the inverse models, and in Figure 8 B the observer is using its own cognitive map. The final applicability levels achieved by each inverse model are shown in Table 3.

The top graph effectively shows the observer's attempt to determine, through simulation, what actions the demonstrator *believes* it can perform; the lower graph is the observer calculating what actions the demonstrator can actually perform, given the state of the world as the observer knows it to be. In the first three episodes, the demonstrator's cognitive map and the observer's own are in agreement as to what inverse models are applicable: the 'nudge object 3' inverse model is not simulated for the demonstrator in the first two episodes as the observer determines that the demonstrator is unaware of object 3's existence (through the perceptual perspective taking). While the demonstrator is looking at object 3, object 1 is moved to an un-nudgeable position; the demonstrator does not see this, but the observer does, the result being that the observer calculates that the demonstrator still believes that touching object 1 is possible, even though it itself knows that the action cannot be accomplished. Upon the demonstrator rotating back to observe objects 1 and 2 in episode 5, the false belief is resolved and the applicability levels are once again in agreement.

5.3 Experiment 3

Experiment 3 took the perspective taking-action simulation of Experiment 2 further, by having the observer use its PTVRs to predict the visual feedback resulting from potential demonstrator actions, and through this, predict the updates to the demonstrator's cognitive map and how this would effect the outcome of each action. Figure 9 shows the results. Figure 9 A shows the applicability levels of the three inverse models over the four episodes, as determined by the observer when observing the demonstrator and basing its action simulations on its representation of the demonstrator's cognitive map. Figure 9 B, C and D show the cognitive map updates predicted by each of the three 'ghosts', as used by the observer during prediction of visual feedback.

In this experiment, the demonstrator robot may not see an object being moved *at the time*, but if it believes an action with that object

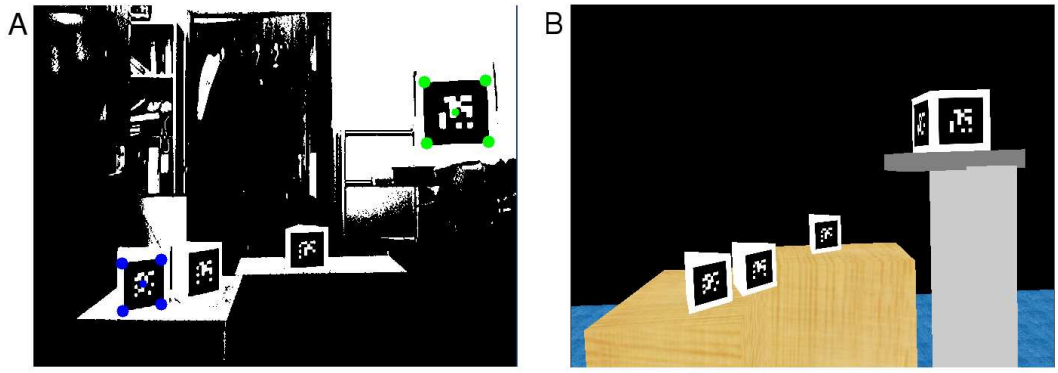


Figure 6. The observer's view of the scene. (A) shows the thresholded camera image sent to the inverse vision models. Objects 1 and 2 are on the table facing the demonstrator, and object 3 is on the table facing the observer. (B) shows the observer's cognitive map, rendered by OpenGL. The three fiducial markers can clearly be seen on the tables, and the demonstrator robot (and its 'hat') can be seen to the right.

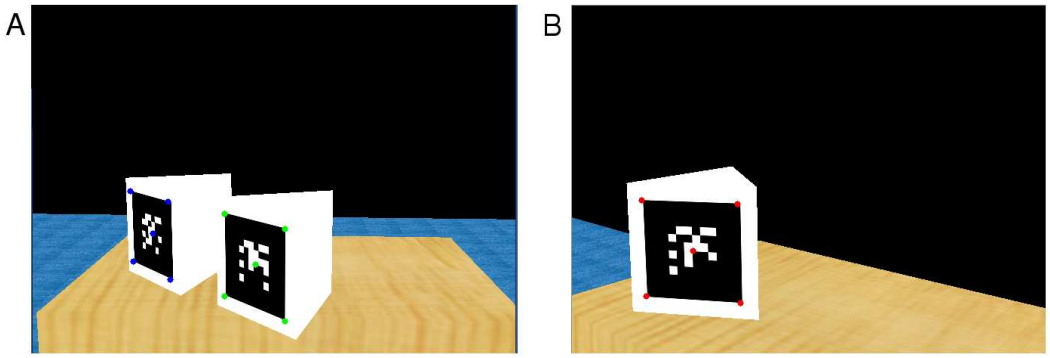


Figure 7. The demonstrator's view of the scene, re-created by the observer in simulation. (A) shows the what the demonstrator sees at the beginning of the experiment, objects 1 and 2. In (B), the demonstrator robot has rotated 45 degrees to the right, and the observer determines that it is able to see object 3. The demonstrator's actual view of (A) is shown in Figure 5.

is still possible and begins to execute it, then after it has rotated and seen the new object configuration, its cognitive map will be updated, the applicability of the action re-calculated, and then it will stop execution since it realises the action is now impossible. Episodes 3 and 4 show this; while the demonstrator is looking at object 3, objects 1 and 2 are moved away from the edge of the table. The demonstrator still believes that the objects are touchable, and so the observer sends out 'ghosts' to simulate how the action may unfold. The spikes in Figure 9 B and C show the predicted updates to the demonstrator's cognitive map when it sees that the objects have moved; as can be seen from figure Figure 9 A, negative applicabilities are calculated and the observer predicts the demonstrator will stop executing those actions. In episode four, the demonstrator rotates to observe the new configuration of objects 1 and 2, and unseen, object 3 is moved away. Figure 9 D shows the resulting cognitive map update for that episode. Again, the result is that the inverse model is no longer applicable and the action is halted mid-execution.

6 Discussion

In developmental psychology, several experimental tasks have been devised in order to investigate the development of cognitive perspective-taking abilities in the paradigm of *false belief*. One of the first tasks in this field was devised by the developmental psychologists Heinz Wimmer and Josef Perner, in response to Daniel Dennett's critique of the experimental protocols used by David Premack and Guy Woodruff in their seminal article that originated the term 'Theory of Mind' [19, 14]. This is the *action prediction* task (also known as the "unexpected transfer" task).

The action prediction task tests an observer determining what a target agent will do when holding a false belief about the world. The test subject, usually a child, observes a puppet-show involving the main character, "Maxi", and his mother. In the show, Maxi watches his mother place a chocolate bar inside a box. Maxi then leaves the room and his mother transfers the chocolate from that box into a different one. Maxi then returns, and the subject is asked where he will look for the chocolate. Further questions include what Maxi would tell to someone he wants to deceive as to the location of the chocolate, and someone he would want to tell the truth to. The result of this task is that four-year-old children give predictions based on correctly attributing the false belief, whereas younger children do not.

Through the use of the cognitive map perspective taking described in this paper, the observer would be able to solve this task. By being able to represent the cognitive map of the demonstrator robot separate to its own, the observer robot is intrinsically able to represent the concept that the demonstrator may possess a false belief about the

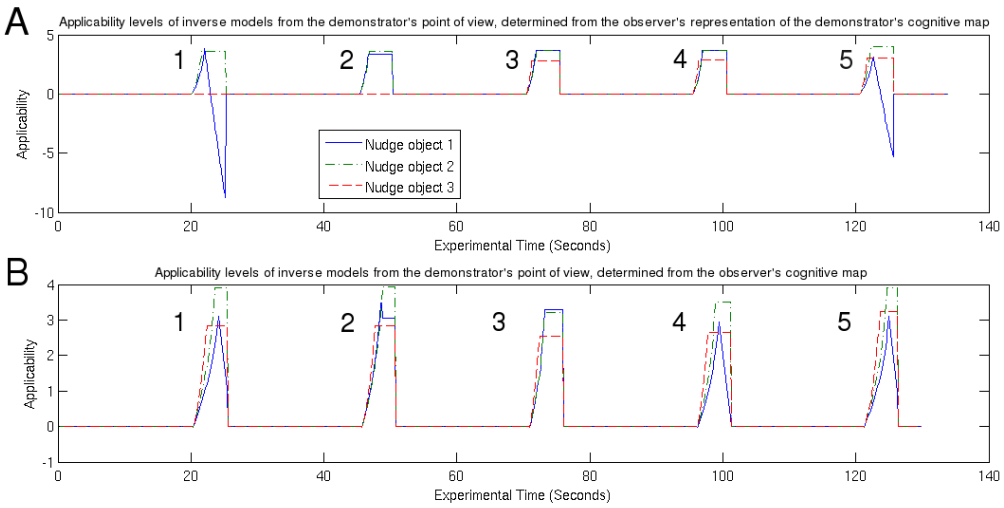


Figure 8. Applicability levels of the observer's inverse models to the demonstrator, over five repeated episodes. Each episode lasted five seconds, after which the applicability levels were reset to zero. Table 3 shows the final applicability levels for each inverse model at the end of the each episode.

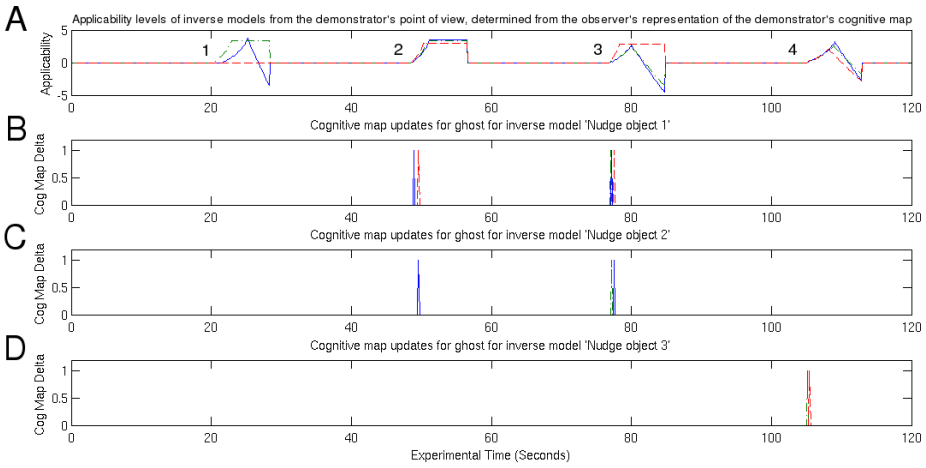


Figure 9. **A.** Applicability levels of the observer's inverse models to the demonstrator, over four repeated episodes. **B, C, D.** Cognitive map updates for each of the three 'ghosts', executing the inverse models 'nudge object' 1, 2, and 3 respectively. A spike indicates that the 'ghost' has seen something that necessitates a change to the cognitive map, and an update is made accordingly. The legend for these graphs is the same as for Figure 8.

location of objects in the world, due to objects moving outside the field-of-view, or object movement being obscured due to occlusions within the field of view. When asked to make predictions as to what the demonstrator may do in such situations, the observer robot is then able to take into account the false belief in the demonstrator's goal setting and action planning. This is illustrated through the results to part 3 of experiment 1, detailed in section 5.1. Through perceptual perspective taking the observer knows that the demonstrator cannot see object 1 being moved — and so, it updates its *own* cognitive map with the change in position, but not the demonstrator's.

Knowledge of this kind, as to the presence of false beliefs in observed agents, can be used by an observer to determine what actions a target agent considers to be available to it, as opposed to what actions it can in fact perform. This information is useful when priming a Simulation-Theory based architecture, such as the HAMMER architecture, with the action simulations it requires for action recognition. The demonstrator will derive its own action goals from what it believes to be the state of the world and move accordingly, and without a representation of the demonstrator's cognitive map, the observer will feed its perspective transform with its own world-state beliefs and potentially end up hypothesizing different goals for its action generation systems — this results in the comparison between internally generated action and observed action being meaningless. In other words, using perceptual perspective taking alone means that we see the world as *we* believe it is from the demonstrator's point of view, whereas what we need to do, in order to infer intention, is see the world as the *demonstrator* believes it is from the demonstrator's point of view. To do the former is to risk not recognising the demonstrator's movements and their action context at all, or to mis-recognise the action as being something else, or to be unable

Table 3. Final applicability levels for each inverse model shown in Figure 8. 'D' indicates that the observer is using its representation of the demonstrator's cognitive map when determining what inverse models are applicable; 'O' indicates that the observer is using its own cognitive map to determine the applicability level of the inverse models. The numbers highlighted in bold type, for 'Nudge object 1' in episode 4, indicates the situation where the observer determines that the demonstrator may possess a *false belief* as to the actions it can make. The absence applicability levels for 'Nudge object 3' in episodes 1 and 2 is due to the demonstrator robot being unaware, at that stage, of the existence of object 3.

	Episode 1		**Episode 2**		**Episode 3**		**Episode 4**		**Episode 5**	
I-Model	D	O	D	O	D	O	D	O	D	O
Nudge object 1	-8.80	1.07	3.35	3.05	3.65	3.28	**3.65**	**0.26**	-5.28	0.98
Nudge object 2	3.59	3.89	3.59	3.93	3.69	3.22	3.69	3.50	3.97	3.90
Nudge object 3	—	2.85	—	2.86	2.83	2.55	2.87	2.65	3.01	3.25

to interpret the demonstrator's goal, and therefore be unable to imitate or learn. The results for experiments 2 and 3 show how through coupling the perspective taking architecture developed in this paper to the action simulation capabilities of HAMMER, the observer can successfully *predict* and *attribute* actions to the demonstrator, while taking into account prior knowledge and experience, and potential false beliefs.

In previous research, the HAMMER architecture was used to model and make predictions regarding the visuomotor 'mirror' neurons found in area F5 of macaque monkey premotor cortex [3]. These neurons are active both when observing an object-directed action, and when performing the same action, leading to suggestions that they underly the imitation capability. Recently, it was found that a subset of these neurons fire even when the object goal of the action is hidden from view, so long as the observer has prior knowledge of the object's presence [17]. With the addition of the cognitive map mechanism described in this paper, HAMMER gains this capability, by keeping a long-term memory of the locations of objects. This can be seen in the results for episode 2 of Experiment 2, where object 1 occludes object 2 from the observer's sight, but the action simulation is still performed. Furthermore, the results of section 5 offer a further prediction—that when a demonstrator performs an action based on a known *false belief* as to the presence of an object, the observer's mirror neurons will fire. Although there is currently no evidence either way, this would lend support to the hypothesis that the mirror neurons encode *intention* and underly action understanding, in addition to action recognition.

7 Conclusions

In this paper we have presented a perspective-taking architecture that uses simulation of visual perception to build up and maintain representations of the cognitive map of a demonstrator. This mechanism, used to improve the state information provided to the HAMMER imitation architecture, was deployed onto robots for perspective-taking and action-prediction experiments, in which an observer successfully attributed potential actions and action predictions to a demonstrator possessing false beliefs regarding the environment. In future work, the mechanism will extended and investigated in experiments involving the observer inferring false beliefs from the actions of a demonstrator.

REFERENCES

[1] E. K. Ackermann, 'Perspective-taking and object construction: Two keys to learning', in *Constructionism in Practice: Designing, Thinking and Learning in a Digital World*, eds., Y. Kafai and M. Resnick, chapter 2, 25–37, Mahwah, New Jersey: Lawrence Erlbaum Associates, first edn., (1996).

[2] M. Billinghurst, H. Kato, and I. Poupyrev, 'The magicbook: A transitional AR interface', *Computers and Graphics*, 745–753, (November 2001).

[3] Y. Demiris, 'Imitation, mirror neurons, and the learning of movement sequences', in *Proceedings of the International Conference on Neural Information Processing (ICONIP-2002)*, pp. 111–115. IEEE Press, (2002).

[4] Y. Demiris and M. Johnson, 'Distributed, predictive perception of actions: A biologically inspired robotics architecture for imitation and learning', *Connection Science*, **15**(4), 231–243, (December 2003).

[5] Y. Demiris and B. Khadhouri, 'Hierarchical attentive multiple models for execution and recognition', *Robotics and Autonomous Systems*, **54**, 361–369, (2006).

[6] A. Drolet, R. Larrick, and M. W. Morris, 'Thinking of others: How perspective taking changes negotiators' aspirations and fairness perceptions as a function of negotiator relationships', *Basic and Applied Social Psychology*, **20**(1), 23–31, (1998).

[7] A. D. Galinsky, G. Ku, and C. S. Wang, 'Perspective-taking and self-other overlap: Fostering social bonds and facilitating social coordination', *Group Processes and Intergroup Relations*, **8**(2), 109–124, (2005).

[8] V. Gallese, 'The manifold nature of interpersonal relations: the quest for a common mechanism', *Phil. Trans. of the Royal Society of London B*, **358**, 517–528, (2003).

[9] R. M. Gordon, 'Simulation vs theory-theory', in *The MIT Encyclopædia of the Cognitive Sciences*, eds., R. A. Wilson and F. Keil, 765–766, MIT Press, (1999).

[10] M. Johnson and Y. Demiris, 'Hierarchies of coupled inverse and forward models for abstraction in robot action planning, recognition and imitation', in *Proceedings of the AISB 2005 "Third International Symposium on Imitation in Animals and Artifacts"*, (April 2005).

[11] M. Johnson and Y. Demiris, 'Perceptual perspective taking and action recognition', *International Journal of Advanced Robotic Systems*, **2**(4), 301–308, (December 2005).

[12] K. S. Narendra and J. Balakrishnan, 'Adaptive control using multiple models', *IEEE Transactions on Automatic Control*, **42**(2), 171–187, (February 1997).

[13] S. Nichols and S. P. Stich, *Mindreading*, Oxford University Press, 2003.

[14] D. Premack and G. Woodruff, 'Does the chimpanzee have a theory of mind?', *Behavioral and Brain Sciences*, **4**, 515–526, (1978).

[15] A. C. Schultz and J. G. Trafton, 'Towards collaboration with robots in shared space: spatial perspective and frames of reference', *Interactions*, 22–23, (March-April 2005).

[16] G. Simmons and Y. Demiris, 'Object grasping using the minimum variance model', *Biological Cybernetics*, **94**(5), 393–407, (May 2006).

[17] M. A. Umilta, E. Kohler, V. Gallese, L. Fogassi, L. Fadiga, C. Keysers, and G. Rizzolatti, 'I know what you are doing: A neurophysiological study', *Neuron*, **31**, 155–165, (July 2001).

[18] Y. Wada and M. Kawato, 'A neural network model for arm trajectory formation using forward and inverse dynamics models', *Neural Networks*, **6**, 919–932, (1993).

[19] H. Wimmer and J. Perner, 'Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception', *Cognition*, **13**, 103–128, (1983).

[20] D. M. Wolpert, K. Doya, and M. Kawato, 'A unifying computational framework for motor control and social interaction', *Phil. Trans. of the Royal Society of London B*, **358**, 593–602, (2003).

[21] D. M. Wolpert and M. Kawato, 'Multiple paired forward and inverse models for motor control', *Neural Networks*, **11**, 1317–1329, (1998).