Achieving Corresponding Effects on Multiple Robotic Platforms: Imitating in Context Using Different Effect Metrics

Aris Alissandrakis, Chrystopher L. Nehaniv, Kerstin Dautenhahn, and Joe Saunders*

*Adaptive Systems Research Group
School of Computer Science, University of Hertfordshire
College Lane, Hatfield, Hertfordshire, AL10 9AB, UK
a.alissandrakis@herts.ac.uk

Abstract

One of the fundamental problems in imitation is the *correspondence problem*, how to map between the actions, states and effects of the model and imitator agents, when the embodiment of the agents is dissimilar. In our approach, the matching is according to different metrics and granularity. This paper presents JABBERWOCKY, a system that uses captured data from a human demonstrator to generate appropriate action commands, addressing the correspondence problem in imitation. Towards a characterization of the *space of effect metrics*, we are exploring absolute/relative angle and displacement aspects and focus on the overall arrangement and trajectory of manipulated objects. Using as an example a captured demonstration from a human, the system produces a correspondence solution given a selection of *effect metrics* and starting from dissimilar initial object positions, producing action commands that are then executed by two imitator target platforms (in simulation) to successfully imitate.

1 Introduction

Imitation is a powerful learning tool that can be used by robotic agents to socially learn new skills and tasks. One of the fundamental problems in imitation is the *correspondence problem*, how to map between the actions, states and effects of the model and imitator agents, matching according to different metrics and granularity, when the embodiment of the agents is dissimilar (Nehaniv and Dautenhahn (1998)). The following statement of the *correspondence problem* (Nehaniv and Dautenhahn (2000, 2001, 2002)) draws attention to the fact that the model and imitator agents may not necessary share the same morphology or may not have the same affordances:

> Given an observed behaviour of the model, which from a given starting state leads the model through a sequence (or hierarchy [or program]) of *sub-goals* in *states*, *action* and/or *effects*, one must find and execute a sequence of actions using one's own (possibly dissimilar) embodiment, which from a corresponding starting state, leads through corresponding sub-goals - in corresponding states, actions, and/or effects, while possibly responding to corresponding events.

In this approach, a solution to the correspondence problem can be used to generate a *recipe* (a loose plan) through which an imitator can map sequences of observed actions of the model agent to its own repertoire of actions as constrained by its own embodiment and by context (Nehaniv and Dautenhahn (2000, 2001, 2002)). Qualitatively different kinds of social learning result from matching different combinations of matching actions, states and effects at different levels of granularity (Nehaniv (2003)). The sub-goals define the granularity to match and vice versa.

Artificial agents that have the ability to imitate may use (perhaps more than one) metric to compare the imitator agent's own actions, states and effects with the model's actions, states and effects, in order to evaluate the imitation attempts and discover corresponding actions that they can perform to achieve a similar behaviour. The choice of metrics used is therefore very important as it will have an impact on the quality and character of the imitation. Many interesting and important aspects of the model behaviour

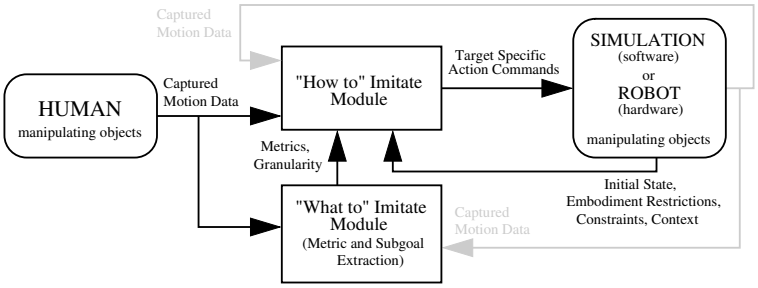

Figure 1: **The JABBERWOCKY system architecture.** Using data captured from a human and given appropriate metrics and sub-goal granularity, the multi-target system can produce action command sequences that when executed by a software or hardware agent can achieve corresponding actions, states and/or effects. The corresponding actions, states and effects as demonstrated by the imitator can also be captured and used as a demonstration for another imitating agent. Differently embodied and constrained target systems in various contexts need to be supported.

need to be considered, as the metrics capture the notion of the salient differences between performed and desired actions and also the difference between attained and desired states and effects (Nehaniv and Dautenhahn (2001, 2002)). The choice of metric determines, in part, *what* will be imitated, whereas solving the correspondence problem concerns *how* to imitate (Dautenhahn and Nehaniv (2002)). In general, aspects of action, state and effect as well as the level of granularity (*what to imitate*) do all play roles in the choice of metric for solving the problem of *how to imitate* (Nehaniv and Dautenhahn (2001); Alissandrakis et al. (2002); Billard et al. (2004)). On-going research is thus addressing the complementary problem of how to *extract sub-goals* and *derive suitable metrics* automatically from observation (Nehaniv and Dautenhahn (2001); Nehaniv (2003); Billard et al. (2004); Calinon and Billard (2004)).

In our setting, it will be desirable to have different kinds of agents in the learning process, i.e. humans and robots interacting socially. Focusing on object manipulation and arrangement demonstrated by a human, this paper presents a system that uses different metrics and granularity to produce action command sequences that when executed by an imitating agent can achieve corresponding effects (manipulandum absolute/relative position, displacement, rotation and orientation). Depending on the particular metrics and granularity used, the corresponding effects will differ (shown in an example), making the appropriate choice of metrics and granularity depend on the task and context.

The work presented in this paper is motivated by the EU Integrated project COGNIRON ("The Cognitive Robot Companion") and addresses the problem of how to teach a robot new complex tasks through human demonstration. The learning algorithms to be developed should be general and address fundamental questions of imitation learning, applied to manipulation tasks. For example a robotic companion at home could acquire knowledge of e.g. laying out a table or drawing on a canvas from observing its human owner. Acquiring such skills socially requires matching different aspects of the effects that the human actions have on objects in the environment. Also the context within which a skill is replicated might require its generalization to various settings and to other types and shapes of objects.

2 The JABBERWOCKY System

This section presents the JABBERWOCKY system developed for the COGNIRON project, that uses captured data from a human demonstrator to generate appropriate action commands (see Figure 1), addressing the correspondence problem in imitation. The action commands can be targeted for various software and hardware platforms. These actions will allow the imitating agent to achieve corresponding actions, states and effects, depending on the given (relevant to the demonstrated task and context) metrics and granularity (provided by a *what to imitate* and *sub-goal extraction* module), embodiment restrictions and constraints (imposed by the targeted imitator platform), and possibly different initial state of the objects in the environment.

The design of the JABBERWOCKY system is informed by the ALICE (**A**ction **L**earning via **I**mitating **C**orresponding **E**mbodiments), a generic framework

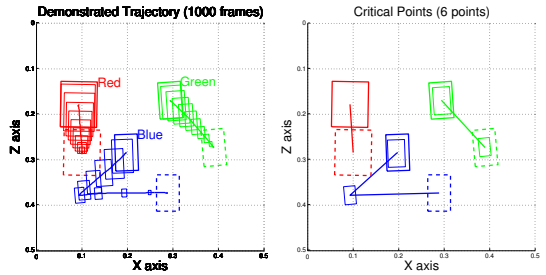

Figure 2: **Captured demonstration (left), and the extracted critical points (right).** The colors (red, green and blue) indicate the three different objects. The dotted outlines indicate the initial position and orientation of the objects, while the solid thick outline the final position. For the demonstration data, the intermediate object's position and orientation is shown with solid thin outlines, linearly scaled (at intervals equal to one tenth of the overall trajectory only, for clarity) to indicate the direction of the movement. For the critical points, each object's position and orientation is shown at every critical point, again linearly scaled.

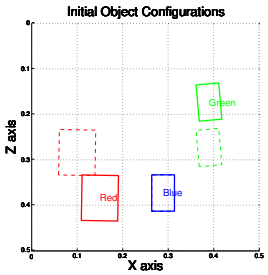

Figure 3: **An example of dissimilar initial object positions.** The dotted outlines indicate the initial position and orientation of the objects in the demonstrator's workspace (from the demonstration shown in Figure 2, left) and the solid outlines the (dissimilar) initial configuration of the objects in the imitator's workspace. The blue object has the same initial position.

for solving the correspondence problem (see Alissandrakis et al. (2002, 2004)). The ALICE framework builds up a library of actions from the repertoire of an imitator agent that can be executed to achieve corresponding actions, states and effects to those of a model agent (according to given metrics and granularity). The ALICE framework provides a functional architecture that informs the design of robotic systems that can learn socially from a human demonstrator.

The system bears some similarity to the one presented in (Kuniyoshi et al. (1994)), but with the main differences that it can use *any* given metric and granularity and that it is designed to be able to generate action commands targeted for a variety of platforms, both in software and hardware to match different behaviour aspects and achieve various types of social learning.

2.1 Demonstrator (Model Agent)

The system uses captured data from a human demonstrator. The demonstrated behaviour is captured using motion sensors (Polhemus LIBERTY^{TM} motion capture system). By attaching the motion sensors on the arms, hands and torso of the human, as well as on the objects that the demonstrator is manipulating, we can obtain the *actions* (e.g. hand movements, gestures), *states* (e.g. arm and body postures) and *effects* (e.g. positioning, displacement, rotation of objects in the workspace) of the demonstrator.

In example shown in Figure 2, the demonstrated task consists of three block objects (colored red, green and blue) arranged in a 2D workspace surface by a human who acts as the demonstrator. The workspace is a square grid 50 cm by 50 cm, and the sizes of the objects are: 10 cm by 8 cm (red) and 8 cm by 5 cm (green and blue). As the manipulations occur only in a 2D plane, only the XZ dimensions are given here (and shown in the figures) omitting the Y dimension (height).

The current work focuses on the *effects* aspect of the demonstrated behaviour, so only the position and orientation of the objects as they are manipulated by the demonstrator are captured, omitting the demonstrator's actions (arm movements) and states (body posture). The choice of initially concentrating on effects for this work is guided by the assumption that the manipulation of objects will be the most important aspect of the demonstrated behaviours that users would like a robotic companion to imitate in a home environment (e.g. fetching objects or arranging them in particular ways).

In ongoing work, three (or more) additional sensors will be used, one attached to the human torso and one at each hand/arm, providing additional information about the demonstrator's states and actions. Taking into account the *states* aspect would help the JABBERWOCKY system solve possible ambiguities when producing the corresponding actions for imitation. For example, a humanoid robot imitator, consider-

ing the states of the demonstrator would obtain possibly useful information e.g. which hand to use (left or right) to reach an object from its current configuration, based on the choice of hand used by the demonstrator.

2.2 *What to Imitate* Module

The character of the imitation will depend on the metrics and granularity (Alissandrakis et al. (2002, 2004); Alissandrakis (2003)). The *what to imitate* module will use the captured demonstration data to extract appropriate sub-goals (granularity) and also discover what metrics must be used to capture the appropriate aspects of the particular demonstration.

In the current implementation of the JABBERWOCKY system the metrics and the sub-goal granularity are given, instead of being discovered by the imitator agents based on the observed demonstrated task. The *what to imitate* module provides a choice of metrics and granularity based on the task and context of the demonstration, although there might not always be a unique, "correct", choice. Here, the various possible metrics and granularity have been selected in advance. It can be shown that the character of the resulting matched effects can be very different, depending on the choice of metrics and granularity used.

The sub-goal granularity is given by finding the *critical points* in the trajectories of the manipulated objects. A critical point occurs when the direction of the captured trajectory and/or the orientation of an object changes by more than a certain threshold.

Several different *effect* metrics have been defined (see section 2.3) that are used in the experiment presented in this paper. In the future the work will be extended to consider also the *state* and *action* aspects of a demonstration.

2.3 Metrics

Towards a characterization of the *space of effect metrics*, we are exploring absolute/relative angle and displacement aspects and focus on the overall arrangement and trajectory of manipulated objects. Looking at how objects can be manipulated (in a non-destructive and combining way), there are two different perspectives: how the object was displaced and how it was rotated. The displacement can be either relative or absolute related to the final position in the workspace, or relative to the other objects within the workspace. The rotation can be also be relative or absolute related to the final orientation of the object. To fully describe the manipulation of an object, both displacement and angular effect aspects must be considered. We consider these aspects in a two-dimensional workspace, such as a table surface.

If the initial configuration of the (same or corresponding) objects is the 'same' for both the model and the imitator agents, then there is no observable distinction between using either the absolute and relative displacement/rotation or the relative position (if the objects are manipulated in the same order). But if the agents are active in a different workspace starting from a different initial configuration of objects, or the timing and the order of the manipulations is not the same, it will be impossible to satisfy simultaneously all the aspects. Therefore choosing to satisfy one particular aspect will result in a qualitatively different effect than if another one was chosen, but still satisfy those similarity quantitative criteria.

2.3.1 Displacement Effect Metrics

The model is moving an object from position X_M to position X'_M on the workspace, achieving an object displacement $\Delta X_M = X'_M - X_M$, where $X_M = \begin{bmatrix} x_M \\ y_M \end{bmatrix}$, $X'_M = \begin{bmatrix} x'_M \\ y'_M \end{bmatrix}$, and $\Delta X_M = X'_M - X_M = \begin{bmatrix} x'_M - x_M \\ y'_M - y_M \end{bmatrix}$. The imitator should move the same (or corresponding) object from position X_I to position X'_I on the workspace, with a displacement $\Delta X_I = X'_I - X_I$, such that a displacement metric is minimised (see Fig. 4, left).

Relative Displacement Effect Metric is minimized if $\Delta X_I = \Delta X_M$ and $X'_I = X_I + \Delta X_M = \begin{bmatrix} x_I \\ y_I \end{bmatrix} + \begin{bmatrix} x'_M - x_M \\ y'_M - y_M \end{bmatrix} = \begin{bmatrix} x_I + x'_M - x_M \\ y_I + y'_M - y_M \end{bmatrix}$.

Absolute Displacement Effect Metric is minimized if $X'_I = X'_M$ and $\Delta X_I = X'_M - X_I = \begin{bmatrix} x'_M - x_I \\ y'_M - y_I \end{bmatrix}$.

Relative Position Effect Metric is minimized if the object is moved to a similar position relative to other objects in the workspace. The *relative position* effect metric is defined here for three objects in the workspace.

The center of the manipulated object is defined as $A = \begin{bmatrix} x_A \\ y_A \end{bmatrix}$, and the centers of the other two objects as $B = \begin{bmatrix} x_B \\ y_B \end{bmatrix}$ and $C = \begin{bmatrix} x_C \\ y_C \end{bmatrix}$. The imitator must move the same (or corresponding) object to form a triangle ABC so that it is the "same" as the triangle formed by the model, i.e. the angles $C\hat{A}B$,

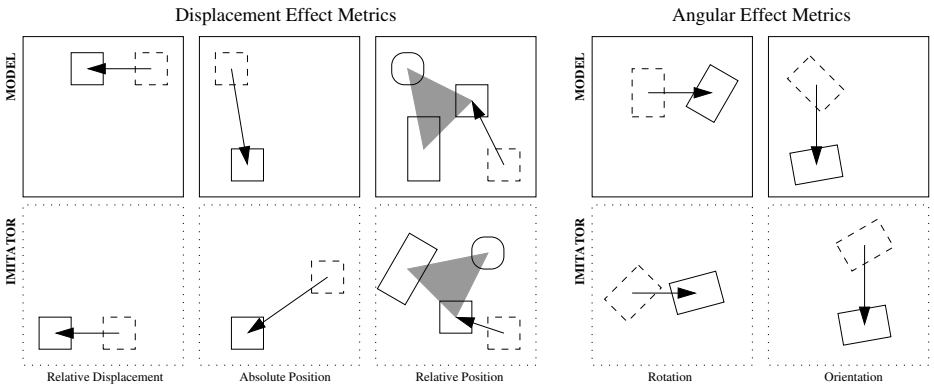

Figure 4: **A selection of *displacement* (left) and *angular* metrics (right).** To evaluate the similarity between object displacements, the *relative displacement*, *absolute position* and *relative position* effect metrics can be used. To evaluate the similarity between object rotations, the *rotation* and *orientation* effect metrics can be used. The second row shows the way the corresponding object (in a different workspace) needs to be moved or rotated by an imitator to match the corresponding effects. The grey triangles are superimposed to show that for the *relative position* effect metric, the relative final positions of the objects are the same.

$A\hat{B}C$ and $B\hat{C}A$ are equal. The triangle sides $\bar{AB}$, $\bar{BC}$ and $\bar{CA}$ can be equal only if the objects start from the same initial configuration for both agents and are manipulated in the same order, so only the equality of the angles can be used in general[1].

The *relative position* effect metric is minimized if $X'_I = A$ and $\Delta X_I = A - X_I = \begin{bmatrix} x_A - x_I \\ y_A - y_I \end{bmatrix}$.

2.3.2 Angular Effect Metrics

The model is rotating an object from orientation θ_M to orientation θ'_M on the workspace, with a rotation $\Delta\theta_M = \theta'_M - \theta_M$. The imitator should rotate the same (or corresponding) object from orientation θ_I to orientation θ'_I on the workspace, with a rotation $\Delta\theta_I = \theta'_I - \theta_I$, such that a displacement metric is minimised (see Fig. 4, right).

Rotation Effect Metric is minimized if $\Delta\theta_I = \Delta\theta_M$ and $\theta'_I = \theta_I + \Delta\theta_M$.

Orientation Effect Metric is minimized if $\theta'_I = \theta'_M$ and $\Delta\theta_I = \theta'_M - \theta_I$.

2.3.3 Other Effect Metrics

Depending on the initial configuration of the corresponding objects in the imitator's workspace, or the particular task that the imitator would like to achieve, it might be desirable to use also other metrics that take into account mirror symmetry, both positional and angular, to features of the environment or other agents. For example:

Mirror Displacement Effect Metric is minimized if $\Delta X_I = -\Delta X_M$ and $X'_I = X_I - \Delta X_M = \begin{bmatrix} x_I \\ y_I \end{bmatrix} - \begin{bmatrix} x'_M - x_M \\ y'_M - y_M \end{bmatrix} = \begin{bmatrix} x_I - x'_M + x_M \\ y_I - y'_M + y_M \end{bmatrix}$.

Mirror Rotation Effect Metric is minimized if $\Delta\theta_I = -\Delta\theta_M$ and $\theta'_I = \theta_I - \Delta\theta_M$.

Parallel Orientation Effect Metric is minimized if $\theta'_I = \vartheta$ and $\Delta\theta_I = \vartheta - \theta_I$, where ϑ is the orientation of a feature in the environment (e.g. one edge of the table). If the features in the workspace of the imitator are the same as the model's, then $\vartheta \equiv \theta'_M$ and this metric becomes equivalent to the *orientation effect metric*.

2.4 Combinations of Effect Metrics

To evaluate both the movement and the orientation of an object, both metric types must be used. To match the observed effect, the (corresponding) object needs to be moved on the workspace according to the displacement given by the displacement effect metric and rotated according to the angular effect metric used.

A weighted combination of more than one displacement metric can also be used, by averaging the displacement vectors that minimise each metric. For example, if $\Delta X_i = \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix}$ is the displacement that minimises a displacement effect metric i, and ω_1, ... , ω_n are the weights of the n displacement effect metrics to be combined, the displacement that

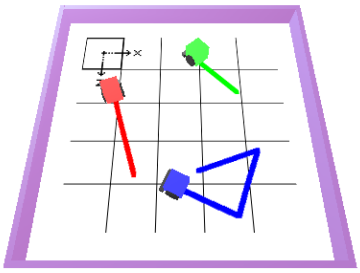

Figure 5: **The *three robots as objects* imitator platform.**

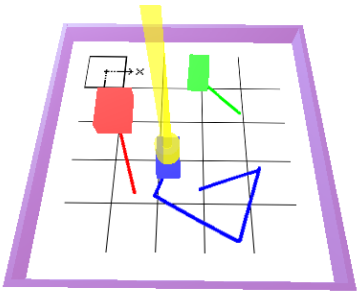

Figure 6: **The *manipulator and three objects* imitator platform.**

minimizes this composite metric is then given by $\Delta X = \begin{bmatrix} |\Delta X| \times cos(\phi) \\ |\Delta X| \times sin(\phi) \end{bmatrix}$, where $|\Delta X| = \omega_1 \times \sqrt{{\Delta x_1}^2 + {\Delta y_1}^2} + ... + \omega_n \times \sqrt{{\Delta x_n}^2 + {\Delta y_n}^2}$ and $\phi = \omega_1 \times tan^{-1}\left(\frac{\Delta y_1}{\Delta x_1}\right) + ... + \omega_n \times tan^{-1}\left(\frac{\Delta y_n}{\Delta x_n}\right)$.

2.5 Imitator

The system is addressing the correspondence problem for dissimilarly embodied imitators, so the *how to imitate* module must produce action commands that can be used by multiple different target platforms as imitator agents, both in simulation (software) and hardware (robots).

Each particular target platform will pose different embodiment restrictions and constraints to the actions, states and effects it can achieve, and eventually to the quality and character of the imitation.

The demonstrator and the imitator might share the same workspace or they might operate in different ones. Even in the same workspace, unless the objects and agents positions are arranged back into the same initial configuration before the imitative behaviour, the context will be different and the imitator therefore has to take that into consideration when imitating.

Two targeted platforms are used in the current realization of the system, both implemented using the Webots™ robot simulation software.

2.5.1 Three Robots As Objects

In the first imitator platform, the imitator's workspace contains no objects. Instead, the imitator is 'embodied' as three mobile robots, each corresponding to one of the objects manipulated by the demonstrator (see Figure 5). Each robot is square 4cm by 4cm (so in this case, besides dissimilar demonstrator-imitator embodiments, there is also dissimilar object correspondence, mapping the objects to mobile robots). The robots can follow the individual trajectories of the objects as arranged by the demonstrator, but cannot match the orientation (while moving) because they are differential wheel robots. Therefore the *angular* effect aspect will be ignored when they imitate, matching only the *displacement* effect aspect.

In the simulation, as the robots move around the workspace, they leave behind a colored trail (of same color as themselves and their corresponding objects) to help visualize the imitated trajectories.

2.5.2 Manipulator and Three Objects

In the second targeted imitator platform, the imitator's workspace contains three objects, of the same size and color as the corresponding objects in the demonstrator's workspace (in this case). The imitator is embodied as a single arm manipulator, positioned above the workspace and able to pick-up, move and rotate the three objects (see Figure 6). This embodiment, although dissimilar to the one of the human demonstrator, is nevertheless able to match both *displacement* and *angular* effect aspects of the demonstration.

As the objects are moved (and rotated) around the workspace by the manipulator in the simulation, they leave behind a colored trail (of same color as themselves) to help visualize the imitated trajectories. The manipulator is shown as a vertical yellow cylinder mounted at the end of a bar positioned above the workspace.

2.6 *How to Imitate* Module

The *how to imitate* module uses the captured data from the demonstration, the metrics and the sub-goal

granularity discovered by the *what to imitate* module to produce a sequence of action commands for an agent to execute and imitate. These action commands are made target specific by taking into account the particular embodiment, affordances and restrictions of the target imitator agent, and also contextual information (including the initial state) for both the agent and the environment. In the current system implementation both the metrics and the sub-goal granularity (critical points) are given.

Concentrating on the *effects* aspect of the demonstrated behaviour to be imitated only, an embodiment-independent solution to the correspondence problem can be found, taking into account the *effect* metrics and the sub-goal granularity. For example consider a human opening a cupboard, removing an object, closing the cupboard and placing the object on a table. This sequence of events can be achieved by agents of varying embodiments, ignoring *state* aspects like e.g. which hand was used to open the cupboard or how the object was held (or grasped) or even *action* aspects e.g. the way the human walked (gait) across the room. Any agent that can open the cupboard, transport the object and place it on the table can potentially imitate the effects of this particular demonstration. But for this solution to be useful to an imitating robotic companion, it must be converted to action commands that take into account its embodiment and also the context (e.g. the cupboard is already open, the object is located on a different shelf in the cupboard, the table is in another room), so that the imitator uses its motors and actuators to achieve the desired effects of the task.

The *how to imitate* module considers the given effect metrics and sub-goal granularity, together with the (possible dissimilar) initial configuration of the objects in the imitator's workspace (also given) to produce initially an embodiment-independent correspondence solution (since only the effects behaviour aspects are considered).

To discover this correspondence, the JABBERWOCKY system currently uses a simple simulation of the 2D workspace that can handle various 'block' objects moving and rotating around, accounting for object collisions and workspace confines. This simulation can replay the captured model data at a given granularity, displaying the trajectory and orientation of the objects as they move and rotate on the workspace, from the initial configuration to the final captured frame. In parallel, starting from a different initial configuration of the same (or different) corresponding objects on the imitator's workspace, the simulation produces a sequence of changes to displacement and rotation for each object, that minimize the given effect metrics.

For example if the effect metric used is the *relative displacement* effect metric, and the demonstrator moved an object 10 cm to the right, then in order to minimize the metric, the corresponding object in the imitator's workspace must be also moved 10 cm to the right. But some displacements or rotations, although minimizing the metric, might be invalid because the path or final position is occupied by other objects or agents, e.g. if the corresponding object is less than 10 cm away from the right edge of the workspace (because the initial position was different), the entire move cannot be performed. The *how to imitate* module will then have to discover an alternative way in the given context (including other agents, static or dynamic obstacles) to achieve the same *effects* according to the metric. In this case it might be acceptable to move the object up to the right edge and then continue the rest of the imitative behaviour. In another context, it might be preferable not to move the object at all. This contextual information should be ideally provided by the *what to imitate* module, based on observations of the currently demonstrated task and not pre-defined. In the current JABBERWOCKY implementation, the system attempts to move (or rotate) the objects until they reach an obstacle (based on simple 2D object collision detection), and then stop, instead of considering another path to reach the position (and/or achieve the orientation) that minimizes (if possible) the metric used.

To imitate and achieve similar *effects* as the model, an imitator agent will have to adopt this (largely) embodiment-independent correspondence solution to move and rotate the objects, using a generated sequence of action command instructions. These action commands will be targeted to multiple imitator platforms, taking into account the embodiment constraints and restrictions of imitator embodiments.

Figure 7 (left) shows an example correspondence converted to action commands for the *three robots as objects* target platform. Each robot is given a sequence of way-points, depending on its corresponding object. For each of these way-points, the robot must use its differential wheel embodiment to move in a straight line up to that position in the workspace, and after reaching the target position, move on to the next. Figure 7 (right) shows the resulting captured imitative behaviour.

Figure 8 (left) shows an example correspondence converted to action commands for the *manipulator and three objects* target platform. The action sequence consists of a continuous (closed) path, with

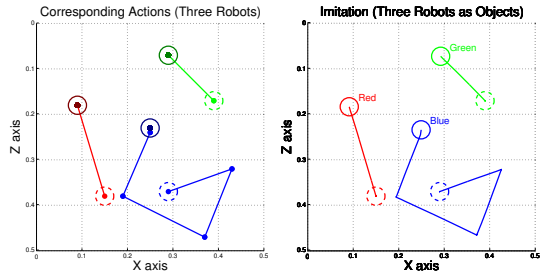

Figure 7: **An example of corresponding action commands for the *three robots as objects* imitator platform (left) and the resulting imitative behaviour (right).** Using the critical points shown in Figure 2, starting from the initial positions shown in Figure 3, and minimizing the *absolute displacement* (red object), *relative displacement* (green) and *relative position* (blue) effect metrics, each of the robots must move along the way-points shown (left). The initial (dotted outline) and final (solid outline) positions are shown as circles, indicating that the orientation of the robots is not considered (the actual robots are square, but of equivalent size). Each way-point is indicated as a dot. The robots then perform an imitative behaviour (in Webots) and the captured results from the simulation are shown in the right plot.

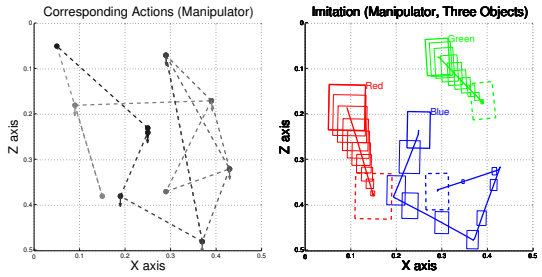

Figure 8: **An example of corresponding action commands for the *manipulator and three objects* imitator platform (left) and the resulting imitative behaviour (right).** Using the critical points shown in Figure 2, starting from the initial positions shown in Figure 3, and minimizing the *absolute displacement* (red object), *relative displacement* (green) and *relative position* (blue) effect metrics, the manipulator must follow the continuous closed path (starting and ending at the left top corner of the workspace) shown as a dotted line (left). Since the human demonstrator did not rotate the objects, no angular effect metrics were used. The line in drawn using a gray to black color gradient to indicate the direction of the path. When reaching an object, the orientation that the object must be rotated to is shown by a small arrow. The manipulator then performs an imitative behaviour (in Webots) and the captured results from the simulation are shown in the right plot.

way-points above the current (and future) positions of the objects. When the manipulator is above an object that must be moved, the manipulator will pick it up, then move (together with the object) to the target position and place the object down (while also, if required, rotating it), before continuing to the next object. To match the effects at each critical point, the order the manipulator approaches the objects is the same (red object, green, blue). If no displacement or rotation is required for an object during each of these turns, that object is ignored, simplifying the manipulator's path. Figure 8 (right) shows the resulting captured imitative behaviour.

3 Conclusions and Discussion.

The experiments shown in Figures 7 and 8 illustrate the diverse character of different successful imitative behaviours optimized to match particular aspects of the effects of demonstrated human manipulation of objects. Aspects captured by metrics for *absolute displacement*, *relative displacement*, *relative position*, *rotation* and *orientation* could all successfully be matched. The results illustrate the multi-platform targetability of the JABBERWOCKY system to map human demonstrated manipulations to matching robotics manipulations (in simulation), generalizing to different initial object configurations.

From the examples shown it becomes apparent that the relative/absolute position and rotation of objects are important aspects of a demonstrated task to match (or not) according to effect metrics, depending on the state of the objects in the environment and the context. The exploratory characterization of the space of effect metrics reveals that matching of "results" is a more sophisticated issue that generally acknowledged. This wide range of possible effect metrics illustrates that even the effect aspect of the correspondence problem for human-robot interaction by itself is already quite complex. Goal extraction in terms of effect metrics and granularity may have many different solutions that might not all be appropriate according to the desired results or context. Depending on the constraints of the imitator embodiment, a 'many-to-one' or 'one-to-many' correspondence between imitator and model sub-goals may be required for specific parts of the task. It is also possible that an

imitating agent has to switch metrics and granularity during the imitation attempt. This has not been emphasized at all in the literature so far (but see Alissandrakis et al. (2004)). This creates particular problems and challenges for sub-goal and metric extraction systems that can be used in programming robots by demonstration. The use of repeated demonstrations (Billard et al. (2004)), saliency detection (Scassellati (1999)) and goal-marking via deixis and non-verbal signaling by humans (Butterworth (2003); Call and Carpenter (2002); Bekkering and Prinz (2002)) may help contribute solutions to these problems. Other research questions yet to be addressed include the importance of order effects in manipulation and establishing object-object correspondence.

Acknowledgements

The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Robot Companion") and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

References

A. Alissandrakis. *Imitation and Solving the Correspondence Problem for Dissimilar Embodiments - A Generic Framework*. PhD thesis, University of Hertfordshire, 2003.

A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn. Imitation with ALICE: Learning to imitate corresponding actions across dissimilar embodiments. *IEEE Trans. Systems, Man & Cybernetics: Part A*, 32(4):482–496, 2002.

A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn. Towards robot cultures? – Learning to imitate in a robotic arm test-bed with dissimilar embodied agents. *Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems*, 5(1):3–44, 2004.

Harold Bekkering and Wolfgang Prinz. Goal representations in imitative actions. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 555–572. MIT Press, 2002.

A. Billard, Y. Epars, S. Calinon, G. Cheng, and S. Schaal. Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, 47:2-3, 2004.

George Butterworth. Pointing is the royal road to language for babies. In Sotaro Kita, editor, *Pointing: Where Language, Culture, and Cognition Meet*, pages 9–26. Lawrence Erlbaum Assoc Inc, 2003.

S. Calinon and A. Billard. Stochastic gesture production and recognition model for a humanoid robot. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, 2004.

Josep Call and Malinda Carpenter. Three sources of information in social learning. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*. MIT Press, 2002.

K. Dautenhahn and C. L. Nehaniv. An agent-based perspective on imitation. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 1–40. MIT Press, 2002.

Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observations of human performance. *IEEE Trans. Robot. Automat.*, 10:799–822, November 1994.

C. L. Nehaniv. Nine billion correspondence problems and some methods for solving them. In *Proc. Second International Symposium on Imitation in Animals and Artifacts – Aberystwyth, Wales, 7-11 April 2003*, pages 93–95. Society for the Study of Artificial Intelligence and Simulation of Behaviour, 2003.

C. L. Nehaniv and K. Dautenhahn. Mapping between dissimilar bodies: Affordances and the algebraic foundations of imitation. In John Demiris and Andreas Birk, editors, *Proceedings European Workshop on Learning Robots 1998 (EWLR-7), Edinburgh, 20 July 1998*, pages 64–72, 1998.

C. L. Nehaniv and K. Dautenhahn. Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. In J. Demiris and A. Birk, editors, *Interdisciplinary Approaches to Robot Learning*, pages 136–161. World Scientific Series in Robotics and Intelligent Systems, 2000.

C. L. Nehaniv and K. Dautenhahn. Like me? - measures of correspondence and imitation. *Cybernetics and Systems*, 32(1-2):11–51, 2001.

C. L. Nehaniv and K. Dautenhahn. The correspondence problem. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 41–61. MIT Press, 2002.

Brian Scassellati. Imitation and mechanisms of joint attention: A developmental structure for building social skills. In C. L. Nehaniv, editor, *Computation for Metaphors, Analogy and Agents*, pages 176–195. Springer Lecture Notes in Artificial Intelligence, Volume 1562, 1999.

Notes

[1] Given $C\hat{A}B_M$, $A\hat{B}C_M$, $B\hat{C}A_M$ and $\overline{BC}_I$, we can find the other two sides $\overline{AC}_I = \sqrt{\frac{(1-cos^2(A\hat{B}C_M))\times\overline{BC}_I^2}{(1-cos^2(C\hat{A}B_M))}}$ and $\overline{AB}_I = \sqrt{\frac{(1-cos^2(B\hat{C}A_M))\times\overline{BC}_I^2}{(1-cos^2(C\hat{A}B_M))}}$, to satisfy the equalities $C\hat{A}B_I = C\hat{A}B_M$, $A\hat{B}C_I = A\hat{B}C_M$ and $B\hat{C}A_I = B\hat{C}A_M$.

Assuming that side $\overline{BC}_I$ lies on the $(0, +\infty)$ x-axis with points $\mathcal{B} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\mathcal{C} = \begin{bmatrix} |\overline{BC}_I| \\ 0 \end{bmatrix}$ corresponding to B_I and C_I, we can then find a point $\mathcal{A} = \begin{bmatrix} \frac{\overline{BC}_I^2 - \overline{AC}_I^2 + \overline{AB}_I^2}{2\times\overline{BC}_I} \\ \frac{\sqrt{(-\overline{BC}_I+\overline{AC}_I-\overline{AB}_I)\times(-\overline{BC}_I-\overline{AC}_I+\overline{AB}_I)\times(-\overline{BC}_I+\overline{AC}_I+\overline{AB}_I)\times(\overline{BC}_I+\overline{AC}_I+\overline{AB}_I)}}{2\times\overline{BC}_I} \end{bmatrix}$ corresponding to A_I, such that the equalities $\overline{\mathcal{AB}} = \overline{AB}_I$, $\overline{\mathcal{BC}} = \overline{BC}_I$ and $\overline{\mathcal{CA}} = \overline{CA}_I$ are satisfied.

To find A_I we need to rotate and translate $\mathcal{A}$ in respect to the actual co-ordinates of $B_I = \begin{bmatrix} x_B \\ y_B \end{bmatrix}$ and $C_I = \begin{bmatrix} x_C \\ y_C \end{bmatrix}$ in the imitator's workspace: $A = \begin{bmatrix} x_A \\ y_A \end{bmatrix} = \begin{bmatrix} cos\phi & sin\phi \\ -sin\phi & cos\phi \end{bmatrix} \times \mathcal{A} + \begin{bmatrix} x_B \\ y_B \end{bmatrix}$, where $\phi = tan^{-1}\left(\frac{y_C - y_B}{x_C - x_B}\right)$.