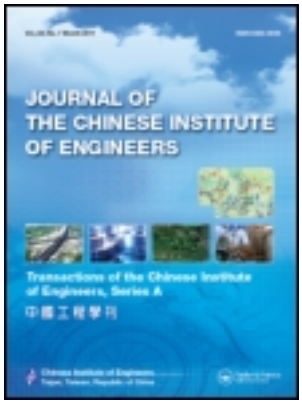

Journal of the Chinese Institute of Engineers
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcie20

Intention deduction from demonstrated trajectory for tool-handling task

Hoa-Yu Chan [a] , Kuu-young Young [b c] & Hsin-Chia Fu [a]

[a] Department of Computer Science , National Chiao-Tung University , 1001 University Road, Hsinchu , Taiwan 300 , ROC

[b] Department of Electrical Engineering , National Chiao-Tung University , 1001 University Road, Hsinchu , Taiwan 300 , ROC

[c] National Chiao-Tung University Vision Research Center , 1001 University Road, Hsinchu , Taiwan 300 , ROC

Published online: 10 Oct 2012.

To cite this article: Hoa-Yu Chan , Kuu-young Young & Hsin-Chia Fu (2013) Intention deduction from demonstrated trajectory for tool-handling task, Journal of the Chinese Institute of Engineers, 36:2, 190-201, DOI: 10.1080/02533839.2012.730249

To link to this article: http://dx.doi.org/10.1080/02533839.2012.730249

Journal of the Chinese Institute of Engineers, 2013
Vol. 36, No. 2, 190–201, http://dx.doi.org/10.1080/02533839.2012.730249

Intention deduction from demonstrated trajectory for tool-handling task

Hoa-Yu Chan[a], Kuu-young Young[bc]* and Hsin-Chia Fu[a]

[a]*Department of Computer Science, National Chiao-Tung University, 1001 University Road, Hsinchu, Taiwan 300, ROC;*
[b]*Department of Electrical Engineering, National Chiao-Tung University, 1001 University Road, Hsinchu, Taiwan 300, ROC;*
[c]*National Chiao-Tung University Vision Research Center, 1001 University Road, Hsinchu, Taiwan 300, ROC*

(*Received 6 May 2011; final version received 7 November 2011*)

When the robot comes to a home-like environment, its programming becomes very demanding. The concept of learning by demonstration is thus introduced, which may remove the load of detailed analysis and programming from the user. Following this concept, in this article, we propose a novel approach for the robot to deduce the intention of the demonstrator from the trajectories during task execution. We focus on the tool-handling task, which is common in the home environment, but complicated for analysis. The proposed approach does not pre-define motions or put constraints on motion speed, while allowing the event order to be altered and allowing for the presence of redundant operations during the demonstration. We apply the concept of cross-validation to locate the portions of the trajectory that correspond to delicate and skillful maneuvering, and apply an algorithm based on dynamic programming previously developed to search for the most probable intention. In experiments, we applied the proposed approach for two different kinds of tasks, the pouring and coffee-making tasks, with the number of objects and their locations varied during demonstrations. To further investigate our method's scalability and generality, we also performed intensive analysis on the parameters involved in the tasks.

Keywords: intention deduction; learning by demonstration; tool-handling task

1. Introduction

Due to progress in service robots, more robots are entering home and office environments. It can be expected that many challenging problems will emerge when they deal with these highly uncertain and varying environments, such as path planning and manipulation, among others (Choset *et al*. 2004, Lu and Hwang 2009). One issue of interest is how to teach the robot to perform daily jobs effectively. To relieve the human operator of detailed task analysis and program coding, researchers have proposed letting the robot learn how to execute the task from observing human demonstration by itself (Kuniyoshi *et al*. 1994). Among them, Calinon *et al*. (2007) proposed an approach using Gaussian mixture regression and Lagrange optimization to extract unchanged motions from multiple demonstrations. The proposed approach demands the order of the operating motions be the same during demonstrations. Dautenhahn and Nehaniv (2002) proposed an approach for the robot to learn from human demonstration by imitation, referred to as the correspondence problem, and later the team developed a system that can learn two-dimensional (2D) arranging tasks (Alissandrakis *et al*. 2005a,b). Pardowitz *et al*. (2006) proposed a hierarchical structure for the robot to deal with complex tasks while the motion order can be changed, and later they went on analyzing human motion features for high-level tasks (Pardowitz *et al*. 2007). With both symbolic and trajectory levels of skill representation, Ogawara *et al*. (2003) proposed a method that determines the essential motions from the possible motions. In this article, we also propose an approach for the robot to learn the human intention from her/his demonstration. To allow the human operator more natural manipulation during demonstration, the proposed approach (a) does not need to pre-define motions; (b) does not constrain the operator to perform the task using a certain motion speed or motion type; (c) allows the order of the events to be altered; and (d) allows some redundant operations.

Pardowitz *et al*. (2007) classify the motions of human manipulations into three different types by their goals: transport operations for moving objects, device handling for changing the internal states of the objects, and tool handling for using tools to interact with objects. We focus on the tool-handling task, which is common in daily life. This kind of task is

*Corresponding author. Email: kyoung@mail.nctu.edu.tw

complicated for analysis, because the tool can operate on multiple objects sequentially without leaving the hand. Based on the concept of cross-validation, but with some modification, we propose an approach to identify the portions of the trajectory corresponding to the delicate and dexterous maneuvers of the demonstrator, referred to as motion features. These motion features, in some sense, exhibit human skills in executing a certain task. The challenge for the proposed approach is how to find the correct intentions, among all possible ones, that lead to the demonstrated trajectories. To tackle the complexity, we apply the method of dynamic programming previously developed for search (Chan *et al*. 2010, 2011). For demonstration, experiments based on two different kinds of tasks, the pouring and coffee-making tasks, were performed. During the experiments, the locations of the operated objects and the operating sequence varied, and the motion features derived from the demonstrated trajectories were used for task execution under different experimental settings. To demonstrate the scalability and generality of the proposed approach further, we performed intensive analysis on the parameters involved in the tasks, such as numbers of objects and demonstrations, among others.

2. Proposed approach

The tool-handling task involves interactions between tools and objects (Ogawara *et al*. 2001). The resultant trajectory from task execution can be divided into two types of motions: delicate motion (D) for delicate maneuver and move motion (M) between the delicate motions (Ogawara *et al*. 2003). The delicate motion is more the focus, since it serves to achieve the goal; by contrast, the move motion is not that critical; while the delicate and move motions are executed alternately. We, thus, take the intention deduction problem to be that of locating the delicate motion from the demonstrated trajectory, and illustrate the proposed approach as follows. In Section 2.1, we describe the process of intention deduction, refined from our previous work (Chan *et al*. 2011). Also, in Section 2.2, we design a series of experiments for investigating its extensibility and robustness.

2.1. *Intention deduction*

Figure 1 shows the conceptual diagram of the proposed approach for intention deduction from demonstration. In Figure 1, the robot first observes a series of human demonstrations and records the corresponding trajectories and environmental states. From these

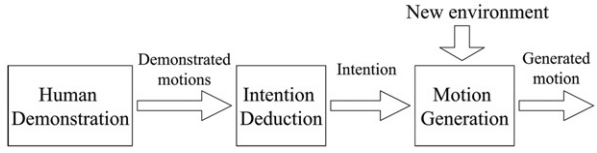

Figure 1. Conceptual diagram of the proposed approach.

recorded motion data, the robot searches for the possible intentions that lead to the delicate motions. The derived intentions can then be used to generate new trajectories that respond to new environmental states. Let us take the pouring task shown in Figure 2, as an example. In Figure 2(a), three vessels A, B, and C are arbitrarily located on the table. Also, in Figure 2(b) and (c), the operator pours the content from vessel A to vessels B and C, respectively, and then places vessel A back on the table. During the demonstrations, the initial locations of the vessels may vary, and so does the pouring sequence. From the recorded trajectories and corresponding locations of the vessels (environmental states), the proposed approach will identify the intention of the operator, i.e., the portions of the trajectory that correspond to the two pouring actions (delicate motions). With the derived intention, the robot is then able to execute the pouring task with the vessels located at various locations and possibly altered pouring sequences.

Before the discussion on the process of intention deduction, we first describe how the motion can be generated under new environmental states when the human intention has already been derived. We start with the representation of the intention I. Assume that there are N delicate motions and S objects involved in a demonstrated task. Because the intention is closely related to the delicate motions of the maneuver, I is formulated as a set of delicate motions, $D_n(t)$, associated with the corresponding objects Obj_s:

$$I = \{D_1(t), D_2(t), \ldots, D_N(t); Obj_1, Obj_2, \ldots, Obj_S\}, \tag{1}$$

where $D_n(t)$ stands for the part of the demonstrated trajectory for delicate motion n and Obj_s the position and orientation of an object s. Note that, because an object may correspond to one, several, or no delicate motions, the number of delicate motions may not be equal to that of the objects. We then introduce the motion index (MI), which serves as an index linking to I. MI is formulated as an ordered set of the time-point pairs, $d_j = \{n_j, l_j, s_j\}$, which provides the starting time n_j, end time l_j, and number of the operated object s_j for each of delicate motions D:

$$MI = \{d_1, d_2, \ldots, d_N\}, \tag{2}$$

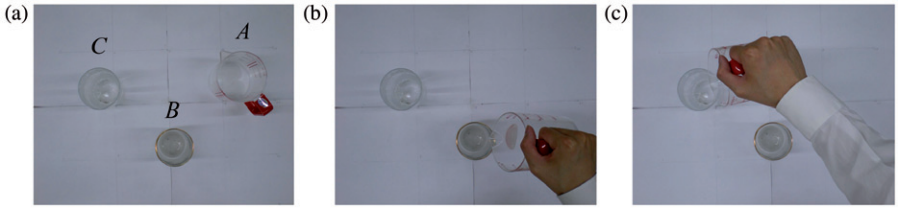

Figure 2. A pouring task: (a) the setting of the vessels; (b) pouring vessel A to vessel B; and (c) pouring vessel A to vessel C.

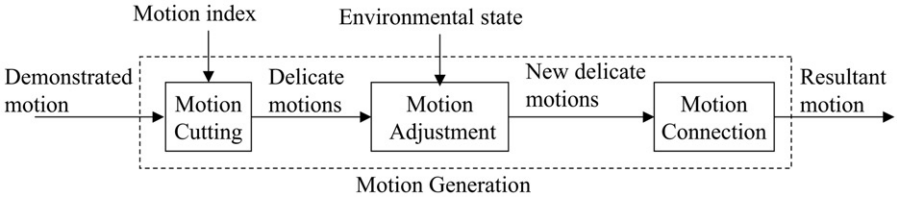

Figure 3. Process for motion generation.

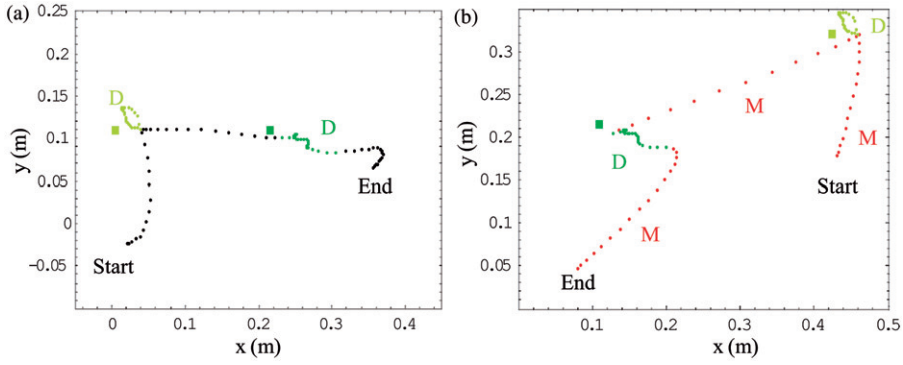

Figure 4. Examples for: (a) motion cutting and (b) motion generation based on the pouring task shown in Figure 2.

With MI representing I, Figure 3 shows the process for motion generation. According to MI, the motion cutting module locates the delicate motions D_j from the demonstrated motion in order. To respond to the new environmental state, the motion adjustment module moves these D_j to match the new locations of the objects and become D_{G_j}. Finally, the motion connection module uses the move motion M_{G_j} to connect every two D_{G_j} smoothly. As its accuracy is not that critical, M_{G_j} is generated using the cubic polynomial. With both D_{G_j} and M_{G_j}, we now have a feasible trajectory Q_G corresponding to the new environmental state:

$$Q_G = \{M_{G_1}, D_{G_1}, M_{G_2}, D_{G_2}, \ldots, D_{G_N}, M_{G_{N+1}}\}. \quad (3)$$

Figure 4(a) shows an example for motion cutting based on the pouring task (Figure 2) and Figure 4(b) that of motion generation. In Figure 4(a), the demonstrated trajectory during task execution is projected on the X–Y plane, where the yellow and green rectangles indicate the locations of vessels B and C. The yellow and green trajectories are the delicate motions determined by the motion cutting module according to the given MI. In Figure 4(b), the yellow and green rectangles indicate the locations of vessels B and C in the new environmental state. In responding to these new locations of vessels B and C, the delicate motions identified in Figure 4(a) are transformed to be the yellow and green trajectories by the motion adjustment module. Finally, the three move motions, the red trajectories, are utilized to connect the two delicate motions smoothly.

From the motion generation process discussed above, we can take the intention deduction process as that of finding proper motion index MI. To find the

optimal *MI* among all *MI* candidates, we introduce first the process for *MI* evaluation, shown in Figure 5. This process evaluates the fitness of the *MI* candidates derived from the demonstrated motion, based on a reasoning that proper *MI* should lead to a generated motion very similar to the human demonstrated motion, which includes all the delicate motions. In Figure 5, from the demonstrated motions, we select one demonstrated motion as the validating motion and the rest as the training motions. We will discuss the selection of validating and training motions later. For an *MI* candidate derived from the validating motion, the motion generation module, described above, generates motions based on the training motions and the environmental state corresponding to the validating motion; the generated motions, with their lengths set to be equal to that of the validating motion, are then compared with the validating motion via the motion comparison module, yielding the differences between them (marked as errors). Because the operator may perform the demonstrations at different speeds and possibly with different orders for the events involved, the corresponding delicate motions are likely to occur at various sampling rates, or to appear in different portions of the demonstrated trajectories. To tackle this, our strategy is to let each of the delicate motions of the validating motion be compared with every portion of the training motion, accompanied by altering sampling rates. Through this comparison process, the generated motion, whose delicate motions lead to the minimum difference when compared with those of the validating motion, is determined as the output and sent to the motion comparison module for the following comparison. As a high search complexity is expected, we come up with an approach analogous to that of dynamic time warping in execution (Sakoe and Chiba 1978).

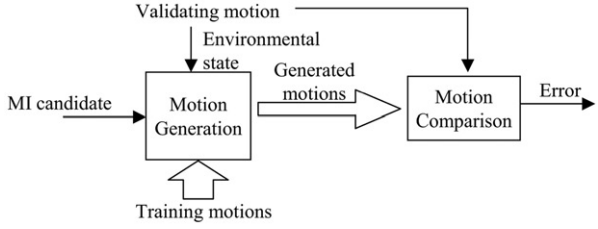

Figure 5. Process for *MI* evaluation.

We go on with the process for *MI* generation, shown in Figure 6, in which, among all the demonstrated motions, one demonstrated motion is first selected as the validating motion, denoted as Q_V, and the rest as the training motions, Q_T, for each sequence of the process. The process will be repeated until each of the demonstrated motions serves as the validating motion once. In the next step, the *MI* generator will locate all possible *MI* candidates from Q_V. Because the proposed approach does not constrain the human operator to perform the task with certain motion speed or motion type, and also allows the order of the events to be altered during demonstration, there is in fact no a priori knowledge for the selection of *MI*. The criterion for *MI* generation is thus to let an *MI* candidate correspond to every portion of Q_V with a duration longer than 0.3 s, as a human cannot cognize an event until 0.3 s after it happens (Sutton *et al.* 1965). It can be expected that there will be a huge number of *MI* candidates. That is why we employ the method of dynamic programming in the search for the optimal *MI*.

With the *MI* evaluation process in Figure 5 and the *MI* generation process in Figure 6, Figure 7 shows the entire process for optimal *MI* derivation. For the outer dotted block in Figure 7, the inputs are the demonstrated motions and each of them serves as the validating motion once. Via the *MI* generation process, *MI* candidates along with the validating and training motions are sent into the *MI* evaluation process to determine which *MI* candidate leads to the minimum error, identified as an optimal *MI* candidate. As each validating motion corresponds to one optimal *MI* candidate, the outputs of the outer dotted block are the

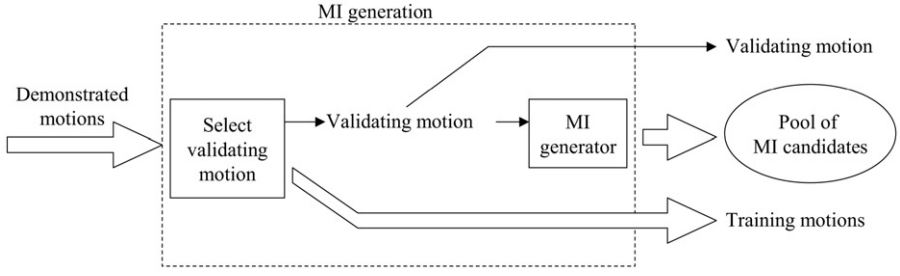

Figure 6. Process for *MI* generation.

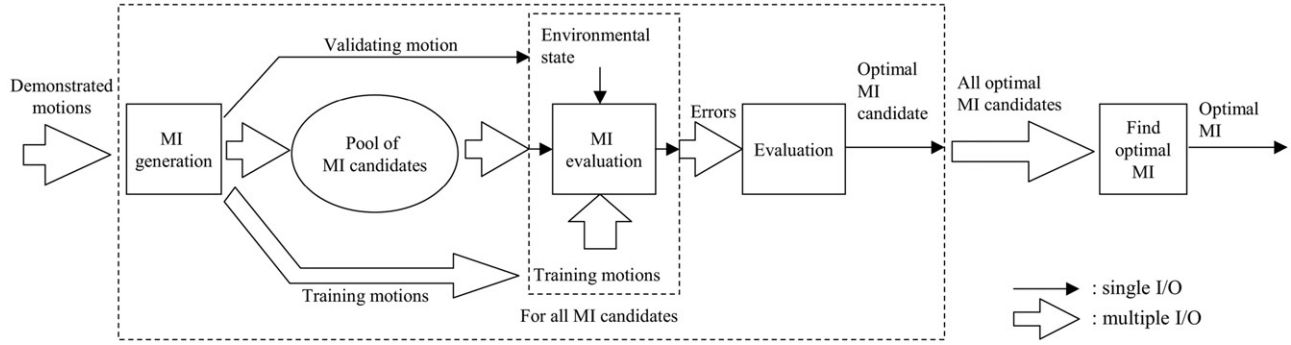

Figure 7. Process for optimal *MI* derivation.

optimal *MI* candidates for each of them. Finally, the optimal *MI* is determined to be the one with the minimum error among all optimal *MI* candidates.

For mathematical formulation of this optimal *MI* derivation process, we start with the description of *MI* for a given validating motion Q_V, denoted as MI_V :

$$MI_V = \{d_{V_1}, d_{V_2}, \ldots, d_{V_N}\}, \tag{4}$$

with

$$d_{V_j} = \{n_{V_j}, l_{V_j}, s_{V_j}\}, \tag{5}$$

where d_{V_j} indexes the delicate motion D_{V_j} with n_{V_j}, l_{V_j}, and s_{V_j} the starting time, end time, and number of the operated object. According to MI_V, Q_V can then be expressed as the combination of a series of delicate and move motions:

$$Q_V = \{M_{V_1}, D_{V_1}, M_{V_2}, D_{V_2}, \ldots, D_{V_N}, M_{V_{N+1}}\}. \tag{6}$$

On the other hand, with the same MI_V, the generated motion Q^i_G for each training motion Q^i_T can be formulated as:

$$Q^i_G = \left\{M^i_{G_1}, D^i_{G_1}, M^i_{G_2}, D^i_{G_2}, \ldots, D^i_{G_N}, M^i_{G_{N+1}}\right\}, \tag{7}$$

where $D^i_{G_j}$ and $M^i_{G_j}$ are its delicate and move motion, respectively. $D^i_{G_j}$ can be determined via the *MI* evaluation process above, in which the minimization between $D^i_{G_j}$ and D_{V_j} is dealt with, and $M^i_{G_j}$ determined by function M_G, which utilizes the cubic polynomial to smoothly connect the two delicate motions, $D^i_{G_{j-1}}$ and $D^i_{G_j}$:

$$M^i_{G_j} = M_G(D^i_{G_{j-1}}, D^i_{G_j}). \tag{8}$$

To determine the optimal motion index MI^*_V, Q_V will be compared with all Q_G generated according to every MI_V. Because we are looking for an MI_V that may induce all the necessary delicate motions, MI^*_V should not induce too much deviation between the delicate motions for Q_V and Q_G, and consequently between the move motions for them. By taking $E_{\max}$ as the maximum difference between the delicate and move motions for Q_V and those Q_G generated for all the training motions corresponding to some MI_V, we determine MI^*_V, among all MI_V, to be the one that leads to the smallest $E_{\max}$:

$$MI^*_V = \arg\min_{MI_V} E_{\max}, \tag{9}$$

$$E_{\max} = \sum_{j=1}^{N} E_D(D_{V_j}) + \sum_{j=1}^{N+1} E_M(D_{V_{j-1}}, D_{V_j}), \tag{10}$$

where

$$E_D(D_V) = \max_i \|D_V - D^i_G\|^2, \tag{11}$$

$$E_M(D_{V_a}, D_{V_b}) = \max_i \|M_V(D_{V_a}, D_{V_b}) - M_G(D^i_{G_a}, D^i_{G_b})\|^2 \tag{12}$$

where E_D computes the difference between the respective delicate motions for Q_V and those Q_G, and E_M that for the move motions, with M_V as a function which outputs the move motion part between two delicate motions of the validating motion, D_{V_a} and D_{V_b}. Because each demonstrated motion serves as the validating motion once, the final optimal motion index MI^{**} for all demonstrated motions will be further chosen as that MI^*, among those for each Q_V, with the smallest corresponding $E_{\max}$, denoted as E^*. As the lengths L_V for $Q_V s$ may not be the same, E^* needs to be normalized before the comparison. MI^{**} is then formulated as:

$$MI^{**}_V = \arg\min_{MI^*_V} E^*/L_V. \tag{13}$$

The search for MI^{**} is of high complexity, as exhibited in Equations (9)–(13). As an attempt to enhance search efficiency, we employ the method of

dynamic programming (Cormen *et al*. 1990) and let the computation of E^* in Equation (13) be expressed in a recursive formulation:

$$E^* = \min_{d_{V_k}} E_R(D_{V_k}) + E_M(D_{V_k}, D_{V_{N+1}}), \tag{14}$$

with

$$E_R(D_{V_k}) = \min_{d_{V_{k-1}}}(E_R(D_{V_{k-1}}) + E_M(D_{V_{k-1}}, D_{V_k})) + E_D(D_{V_k}), \tag{15}$$

where $E_R(D_{V_k})$ stands for the minimum difference between the motions from the first move motion to a given delicate motion; d_{V_k} and $d_{V_{k-1}}$, described in Equation (5), index the delicate motions D_{V_k} and $D_{V_{k-1}}$; and $1 \le k \le N$. Because the number of delicate motions is not known in advance, N and k are not specific numbers. Also, note that the first move motion is generated between D_{V_0} and D_{V_1}, and the last one between D_{V_N} and $D_{V_{N+1}}$, with D_{V_0} and $D_{V_{N+1}}$ taken as the first and last points of the trajectory, respectively. In Equation (14), E^* is derived as the minimum one for all $E_R(D_{V_k})$ with $E_R(D_{V_k})$ computed recursively via Equation (15). With Equations (14) and (15), dynamic programming can take advantage of the table generated for $E_R(D_{V_k})$ to simplify the computation in deriving E^*.

Time complexity for this optimal *MI* derivation process is related to the number (R) and length (L_V) of the demonstrated motions and the number (S) of objects involved in the task. Here, the lengths of the demonstrated motions are assumed to be close. In Equations (14) and (15), the generation of the table for $E_R(D_{V_k})$ takes up most of the time consumed. The table has $O(L_V^2 \cdot S)$ elements, and each element deals with the complexity of the order of $O(R \cdot L_V^3 \cdot S)$. During the entire process, the table needs to be generated R times. The final time complexity is thus computed to be on the order of $O(R^2 \cdot L_V^5 \cdot S^2)$.

Based on the discussions above, the algorithm for intention derivation algorithm is formulated as follows:

Algorithm for intention derivation: Find the intention of the task through R demonstrations.

Step 1: Record the demonstrated trajectories for the R demonstrations. Denote the recorded trajectory for the ith demonstration as Q_i. Set $i = 1$.

Step 2: Select Q_i among the R recorded trajectories as the validating motion Q_V and the rest as the training demonstrations Q_T.

Step 3: Apply the method of dynamic programming, based on Equation (9), to determine the optimal MI^* for Q_V. Let $i = i + 1$. If $i \le R$, go to Step 2; otherwise, go to Step 4.

Step 4: Utilize Equation (13) to determine the optimal MI^{**} for the demonstrator among those MI^* for the R validating motions. MI^{**} is now ready to be used for executing the task under new environmental states.

2.2. *Experimental design for extensibility and robustness*

The proposed approach is developed for general tool-handling tasks, with the appealing features that: (a) there is no need for pre-defined motions, (b) there are no constraints on motion speed or motion type, (c) there is allowance for event-order altering, and (d) allowance is made for redundant operations during demonstration. To investigate its extensibility and robustness, we designed a series of experiments based on two different kinds of tasks: the pouring and coffee-making tasks.

In the pouring task, the operator is asked to hold a vessel and pour the content into other vessel(s), as described in the example shown in Figure 2. The experiments were designed to evaluate the influence from the following factors:

- pouring order during execution;
- number of vessels to pour; and
- number of demonstrations used for *MI* derivation.

We also analyzed the time complexity during task execution, which is expected to match that predicted by the algorithm for intention derivation in Section 2.1. In the coffee-making task, the operator uses a spoon to scoop coffee powder, sugar, or milk from the jars into the coffee cup and stir. The number of jars is fixed for the experiments, but the operator can access the same jar(s) one or several times. In addition to the performance on this coffee-making task, the effect of number of demonstrations on *MI* derivation and time complexity during task execution were also evaluated. Meanwhile, we also analyzed how the presence of the redundant motions affects system performance.

3. Experiment

During the experiments, we applied the proposed approach for the pouring task shown in Figure 8 first. The experiment was divided into two stages: (a) human demonstration and (b) robot execution. Figure 8(a) shows the experimental setup for human demonstration, which includes the human operator and the electromagnetic motion tracking system

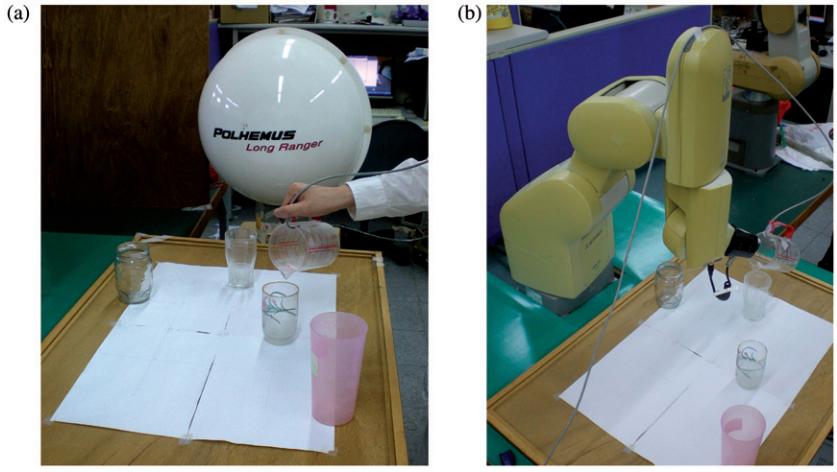

Figure 8. Experimental setups for the pouring task: (a) human demonstration and (b) robot execution.

(FASTRAK, manufactured by Polhemus, USA). There were five vessels randomly placed on the table. The human operator held a vessel (vessel A) and poured the content into the other vessels (vessels B, C, D, and E) on the table. The Polhemus FASTRAK tracking system, with a sampling rate of 30 Hz for each of the sensors, was used to measure and record the demonstrated trajectories and positions of the objects. These trajectories were recorded as 7-D sequences, which consist of positions and orientations (in the form of quaternion) in three and four dimensions, respectively, with the position normalized by its standard deviation. From these recorded trajectories, we applied the intention derivation algorithm in Section 2.1 to derive the intention of the human operator from all possible intentions. We then moved on to the second stage of the experiment, and let the Mitsubishi RV-2A 6-DOF robot manipulator follow the derived intention to execute the pouring task under new environmental states, as shown in Figure 8(b). The proposed approach was programmed mainly using C language, executed in a PC with an Intel E6300 CPU, running at 1.86 GHz with 3.62 Gbyte RAM.

The pouring order for task execution could be the same or arbitrary, and the vessels to pour into were $\{B, C\}$, $\{B, C, D\}$, and $\{B, C, D, E\}$, respectively, leading to six combinations. For each combination, the human operator demonstrated 18 times, and the total number of demonstrations was 108. The locations of the vessels varied during the demonstrations. For each of the 6 combinations, 8 from the 18 demonstrations were randomly chosen as the training data for the intention derivation algorithm to derive the optimal intention, and each one of the remaining 10 demonstrations, which was with a pouring order the same as that corresponding to the optimal intention, chosen as the testing data to evaluate the performance of the derived optimal intention. We performed the intention derivation and evaluation processes five times. Figure 9 shows a typical result for intention derivation for a four-vessel pouring task, in which $\{B, C, D, E\}$ were the vessels to pour into with an arbitrary pouring order. In Figure 9, delicate motions related to vessels B, C, D, and E were identified from the trajectory of vessel A, marked by the yellow, green, blue, and purple blocks, respectively. It was observed that the delicate motions were located at those portions with evident tilt-angle changes, implicating the pouring action. The derived intention for demonstration 3 was determined to be optimal among all. We then let the robot manipulator follow this derived optimal intention to execute the pouring task, in which the vessels were placed in new locations. Figure 10 shows the trajectories generated by the robot manipulator and human operator. Figure 10(a) shows the trajectories of vessels A in the X–Y plane, where color rectangles of yellow, green, blue, and purple indicate vessels $\{B, C, D, E\}$, respectively,

Table 1. Average position error between the trajectories of human operator and robot manipulator.

Vessels	Pouring order	Error
2	Same	0.052
	Arbitrary	0.051
3	Same	0.049
	Arbitrary	0.049
4	Same	0.052
	Arbitrary	0.045

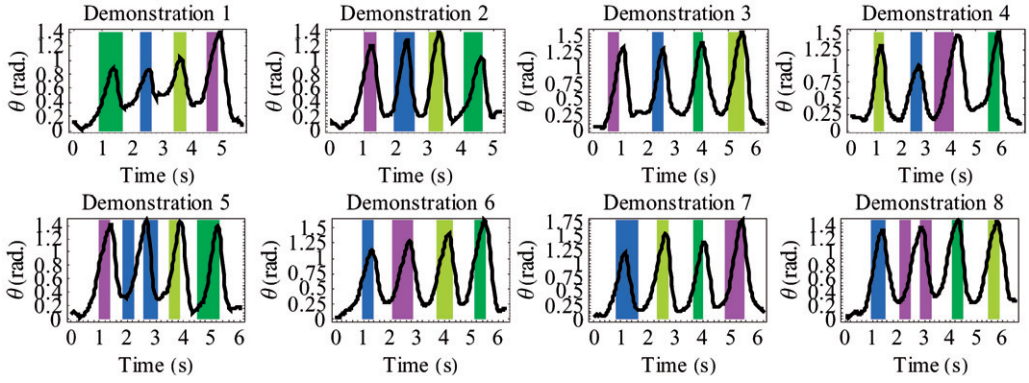

Figure 9. Derived intentions for the four-vessel pouring task.

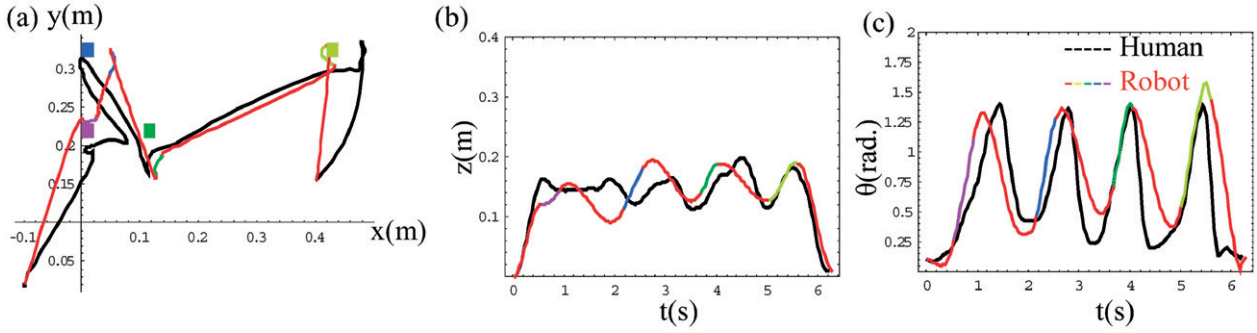

Figure 10. Experimental results for the four-vessel pouring task executed by both the human operator and robot manipulator under new environmental states: (a) trajectory in the X–Y plane; (b) variation of the height; and (c) variation of the tilt angle of vessel A.

Figure 10(b) its variations of the height, and Figure 10(c) its variations of the tilt angle. In Figure 10, the black line is used for the human operator, and the color lines for the robot manipulator (red for move motions and others for delicate motions). In Figure 10(a), the trajectory of the robot manipulator basically followed that of the human operator. Even with some deviations, the robot manipulator still managed to accomplish the pouring task.

To further demonstrate the performance of the proposed approach, Table 1 lists the average position errors between the trajectories of the human operator and robot manipulator for the six combinations of the pouring task. The average position errors were computed based on the results from the five-time intention derivation and evaluation processes performed for each combination, described above. From the closeness of the errors presented in Table 1, the number of vessels and pouring order did not have much effect on the proposed approach. We then analyzed the effect of the pouring order. Table 2 lists the average path length and demonstrating time for task demonstration with the arbitrary and same pouring orders. It was observed that the demonstrations with the arbitrary order led to smaller average path length and shorter demonstrating time, when the number of vessels was above two. It might be because the human operator could conduct the demonstrations more naturally. A short and fast demonstration also implies fewer samples in the motion, thus alleviating the computational load. We also investigated how the number of demonstrations

Table 2. Average path length and demonstrating time for the two pouring orders.

Vessels	Pouring order	Path length (m)	Demonstrating time (s)
2	Same	1.017	3.450
	Arbitrary	0.973	3.452
3	Same	1.436	4.900
	Arbitrary	1.320	4.569
4	Same	1.776	6.396
	Arbitrary	1.523	5.856

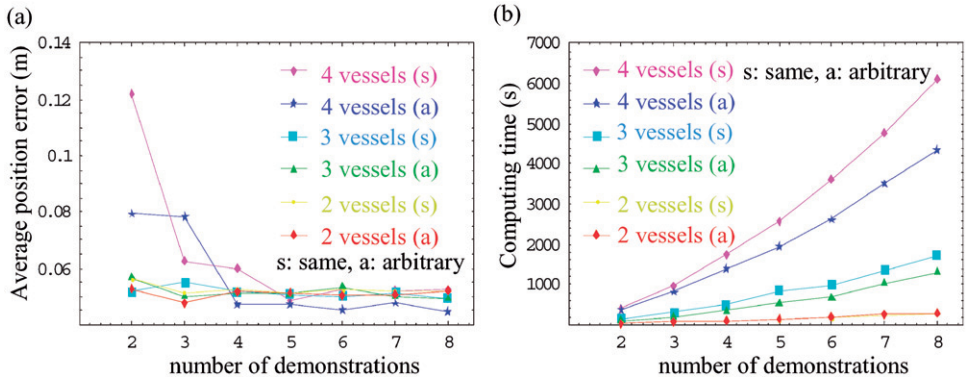

Figure 11. The effect of the number of demonstrations for the pouring task: (a) average position error vs. number of demonstrations and (b) computing time vs. number of demonstrations.

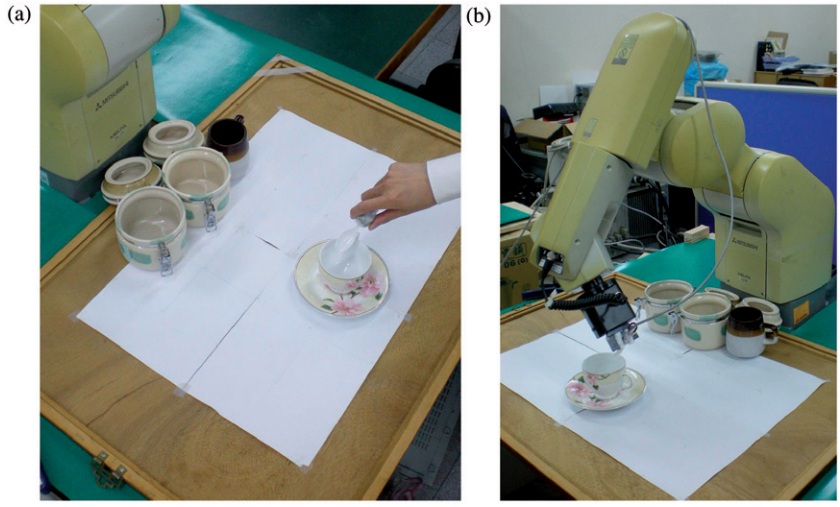

Figure 12. Experimental setups for the coffee-making task: (a) human demonstration and (b) robot execution.

affected the performance of the proposed approach. Figure 11(a) shows the relationship between the average position error and number of demonstrations, where the number of demonstrations increased from two to eight. From the observation on Figure 11(a), the proposed approach demanded at least five demonstrations for evident error reduction for the 4-vessels pouring task, while two demonstrations was enough for the two- and three-vessel cases. This implies that the increase of number of vessels did raise the complexity for the proposed approach to tackle. As for the aspect of computational load, Figure 11(b) shows the relationship between the computing time and number of demonstrations. In Figure 11(b), the computing time approximately increases in line with the square of number of demonstrations, which matches the predicted time complexity discussed in Section 2.1.

To evaluate its extensibility, we then applied the proposed approach for a coffee-making task shown in Figure 12. For this coffee-making task, we employed a spoon *A*, a coffee cup *B*, and jars *C*, *D*, and *E*, containing coffee powder, sugar, and cream, respectively. The location of cup *B* might vary, while those of jars *B*, *C*, and *D* were fixed. During task execution, the human operator held spoon *A* to scoop the coffee powder, sugar, and cream once from jars *C*, *D*, and *E*, respectively, into cup *B*, and then stir. The human operator demonstrated the task 22 times. Using the same procedure executed in the pouring task above, we randomly selected eight demonstrations as the training data for intention derivation and performed the intention derivation and evaluation processes five times. We had also conducted the task for cases without involving the scooping on jar *D* or *E*, indicating no sugar or cream was used in coffee

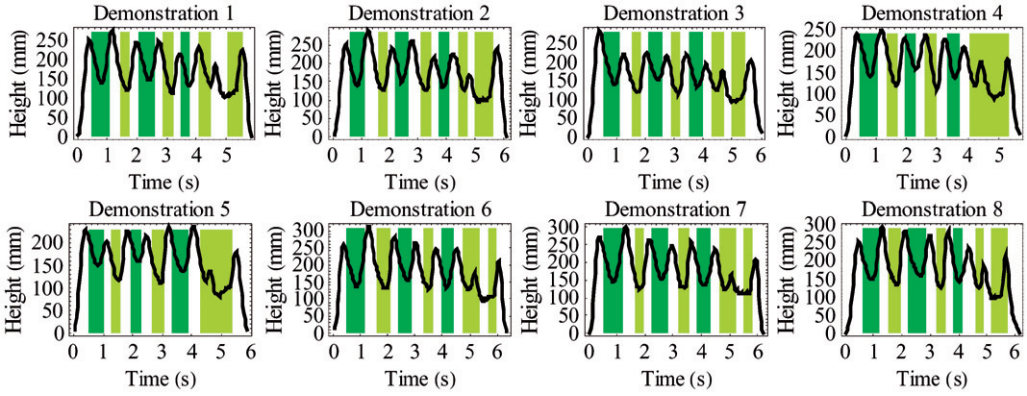

Figure 13. Derived intentions for the coffee-making task.

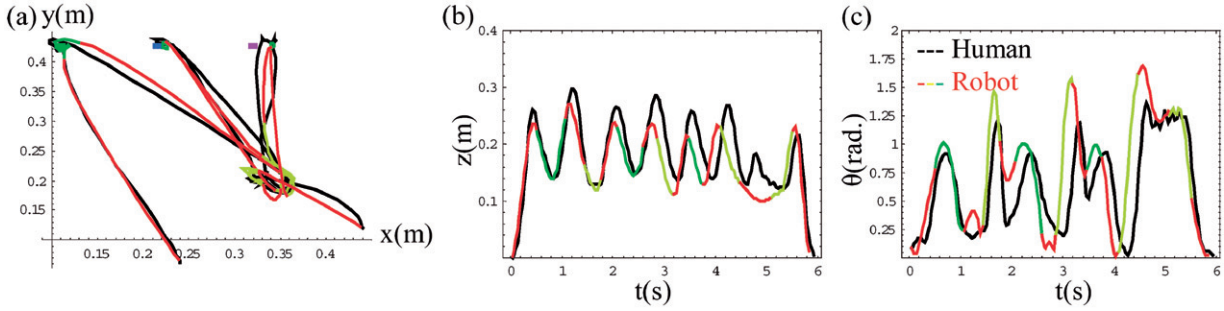

Figure 14. Experimental results for the coffee-making task executed by both the human operator and robot manipulator under new environmental states: (a) trajectory in the X–Y plane; (b) variation of the height; and (c) variation of the tilt angle of spoon A.

making, and also cases involving multiple scooping on the coffee powder, sugar, or cream. Experimental results show that the proposed approach handled all these cases well.

Figure 13 shows a typical result for intention derivation for the coffee-making task involving the scooping of coffee powder, sugar, and cream once. In Figure 13, delicate motions related to jars C, D, and E and cup B were identified from the trajectory of spoon A, marked by two kinds of color blocks (green for jars C, D, and E, and yellow for cup B). The derived intention for demonstration 2 was determined to be optimal, and used by the robot manipulator to execute the task again with cup B in a new location. Figure 14 shows the trajectories generated by the robot manipulator and human operator. Figure 14(a) shows the trajectories of spoon A in the X–Y plane, where color rectangles of yellow, green, blue, and purple indicate cup B and jars C, D, and E, respectively, Figure 14(b) shows variations of height, and Figure 14(c) variations of tilt angle. In Figure 14, the black line is used for the human operator, and the color lines for the robot manipulator (red for move motions and others for delicate motions). In Figure 14(a), the trajectory of the robot manipulator followed that of the human operator to a certain extent. Meanwhile, the robot manipulator accomplished this coffee-making task successfully. Figure 15(a) shows the relationship between the average position error and number of demonstrations, implicating the proposed approach might need at least four demonstrations for evident error reduction. Also, Figure 15(b) shows the relationship between the computing time and number of demonstrations, which also yields that the computing time approximately increases in line with the square of the number of demonstrations.

For further evaluation on its robustness, we analyzed how the presence of redundant operations during demonstration may affect system performance. The analysis was based on the two-vessel pouring task. Among a group of two-vessel demonstrations, we gradually added in some demonstrations involving three or four vessels, taken as the introduction of the redundant operations. With this, we attempted to find

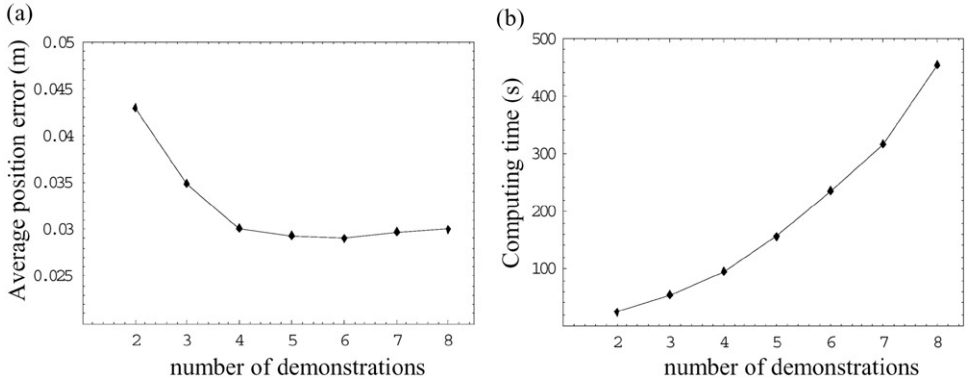

Figure 15. The effect of the number of demonstrations for the coffee-making task: (a) average position error vs. number of demonstrations and (b) computing time vs. number of demonstrations.

Table 3. Number of success in the presence of redundant operations.

2 Vessels	3 Vessels	4 Vessels	Number of success
7	1	0	9
6	1	1	9
5	2	1	8
4	2	2	8

out whether the proposed approach could still successfully recognize that the demonstrations were intended for the two-vessel pouring task, even with some proportion of the demonstrations involving redundant operations. Table 3 lists the number of successes out of 10 tests where the number of demonstrations involving 3 or 4 vessels increased from 1 to 4 out of a total of 8 demonstrations. The proposed approach still reached quite high a success rate at 80%, when the demonstrations involving redundant operations consisted of half of the total demonstrations, since they usually led to larger $E_{\max}$ in deriving the optimal MI, formulated in Equation (9).

4. Conclusion

In this article, we have proposed a novel approach for intention deduction from demonstrated trajectories for tool-handling tasks. The proposed approach does not demand pre-specified motions or put constraints on motion speed during demonstration, and it allows the event order to be altered and the presence of redundant operations. The demonstration can thus be executed in a natural and effective manner. In realization, the concept of cross-validation and the algorithm based on dynamic programming have been employed to search for the optimal intention. We have performed a series of experiments and analyses to demonstrate extensibility and robustness based on both the pouring and coffee-making tasks. In future work, we will apply the proposed approach for various types of tasks related to home-like environments.

Acknowledgments

This study was supported in part by the National Science Council under grant NSC 96-2628-E-009-164-MY3, and also Department of Industrial Technology under grant 97-EC-17-A-02-S1-032. Part of this article has been presented at SICE Annual Conference 2010.

Nomenclature

D	delicate motion
E	the difference between generated motion and validating motion
G	generated motion
i	an index of training motions or generated motions
j, k	an index of delicate motions or move motions
M	move motion
MI	an index linking to a set of delicate motions
Q	a motion consists of delicate motions and move motions
T	training motion
V	validating motion

References

Alissandrakis, A., *et al.*, 2005a. Using JABBERWOCKY to achieve corresponding effects: imitating in context across multiple platforms. *In*: *Workshop on the social mechanisms*

of robot programming by demonstration. *IEEE international conference on robotics and automation*, 22 April 2005, Spain, 9–12.

Alissandrakis, A., *et al*., 2005b. Achieving corresponding effects on multiple robotic platforms: imitating using different effect metrics. *In*: *International symposium on imitation in animals and artifacts*, 12–14 April 2005, UK, 10–19.

Calinon, S., Guenter, F., and Billard, A., 2007. On learning, representing and generalizing a task in a humanoid robot. *IEEE transactions on systems, man and cybernetics – part B: cybernetics*, 37 (2), 286–298.

Chan, H.Y., Young, K.Y., and Fu, H.C., 2010. Learning by demonstration for tool-handling task. *In*: *SICE annual conference*, 18–21 August 2010, Taiwan, 459–464.

Chan, H.Y., Young, K.Y., and Fu, H.C., 2011. Intention learning from human demonstration. *Journal of information science and engineering*, 27 (3), 1123–1136.

Choset, H., *et al*., 2004. *Principles of robot motion*. Cambridge, MA: MIT Press.

Cormen, T.H., Leiserson, C.E., and Rivest, R.L., 1990. *Introduction to algorithms*. Cambridge, MA: MIT Press.

Dautenhahn, K. and Nehaniv, C.L., eds., 2002. The correspondence problem. *In*: *Imitation in animals and artifacts*. Cambridge, MA: MIT Press, 41–61.

Kuniyoshi, Y., Inaba, M., and Inoue, H., 1994. Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE transactions on robotics and automation*, 10 (6), 799–822.

Lu, Y.S. and Hwang, C.S., 2009. High-order variable-structure disturbance estimators with applications to harmonic drive systems. *Journal of the Chinese institute of engineers*, 32 (5), 613–625.

Ogawara, K., *et al*., 2001. Extraction of fine motion through multiple observations of human demonstration by DP matching and combined template matching. *In*: *Workshop on robot and human interactive communication, IEEE international conference*, 18–21 September 2001, Roman, 8–13.

Ogawara, K., *et al*., 2003. Extraction of essential interactions through multiple observations of human demonstrations. *IEEE transactions on industrial electronics*, 50 (4), 667–675.

Pardowitz, M., *et al*., 2007. Incremental learning of tasks from user demonstrations, past experiences and vocal comments. *IEEE transactions on systems, man and cybernetics – part B: cybernetics*, 37 (2), 322–332.

Pardowitz, M., Zoellner, R., and Dillmann, R., 2006. Incremental learning of task sequences with information-theoretic metrics. *In*: *First European robotics symposium*, 16–18 March 2006, Palermo, Italy, 51–63.

Sakoe, H. and Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26 (1), 43–49.

Sutton, S., *et al*., 1965. Evoked potential correlates of stimulus uncertainty. *Science*, 150 (3700), 1187–1188.