Incremental Learning and Memory Consolidation of Whole Body Motion Patterns

Dana Kulić and Yoshihiko Nakamura
Department of Mechano-Informatics, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan
{dana,nakamura}@ynl.t.u-tokyo.ac.jp

Abstract

The ability to learn during continuous and on-line observation would be advantageous for humanoid robots, as it would enable them to learn during co-location and interaction in the human environment. However, when motions are being learned and clustered on-line, there is a tradeoff between classification accuracy and the number of training examples, resulting in potential misclassifications both at the motion and hierarchy formation level. This paper presents an approach enabling fast on-line incremental learning, combined with an incremental memory consolidation process correcting initial misclassifications and errors in organization, to improve the stability and accuracy of the learned motions, analogous to the memory consolidation process following motor learning observed in humans. Following initial organization, motions are randomly selected for re-classification, at both low and high levels of the hierarchy. If a better reclassification is found, the knowledge structure is re-organized to comply. The approach is validated during incremental acquisition of a motion database containing a variety of full body motions.

1. Introduction

As robots move to human environments, the ability to learn and imitate from observing human behavior will become important. This area of research has received increasing attention (Breazeal and Scassellati, 2002, Schaal et al., 2003). However, many of the approaches proposed thus far consider the case where there is an off-line initial learning stage, followed by a static execution and recognition stage. In this case, the number of motions to be learned can be specified a-priori, and the designer can ensure that each motion example is correctly labeled to the appropriate motion group. However, to ensure adaptability to its changing environment and interaction partners, a robot should be capable of continuous learning over its entire lifespan. We have been working towards algorithms that enable the robot to observe, segment and classify demonstrated actions on-line (Kulić et al., 2007a, Kulić et al., 2008b) during co-location and interaction with the (human) teacher. During this type of learning, the number of motion primitives is not known in advance and may be continuously growing, and must be determined autonomously by the robot, as it is observing the motions.

In the proposed approach (Kulić et al., 2007a, Kulić et al., 2008b), a hierarchical tree structure is incrementally formed representing the motions learned by the robot. Each node in the tree represents a motion primitive, which can be used to recognize a similar motion, and also to generate the corresponding motion for the robot. Due to the fact that motions are being sorted incrementally, the algorithm may produce errors as compared to off-line clustering and organization. Two types of errors are possible: errors in classifying individual motions, and errors in the structure formation of the motion hierarchy. However, it can still be advantageous for the robot to quickly learn a rough model of a new motion, rather than waiting for a large number of examples to become available, as the learned motion may then be further refined through other learning modalities, such as practice (Bentivegna et al., 2006) and feedback from the teacher (Nicolescu and Matarić, 2005), which may be more effective than repeated observation alone.

In this paper, we propose a mechanism for incremental, on-line correction of initial clustering and organization errors during on-line motion observation and learning. Our approach is inspired by recent biological studies showing that the structure of motion memory changes following acquisition, in a process termed *memory consolidation* (Shadmehr and Holcomb, 1997, Krakauer and Shadmehr, 2006). Following initial model formation, the model is changed over time to stabilize and improve the initial representation.

1.1 Related Work

Robot skill learning from observation is a longstanding area of research. (Breazeal and Scassellati, 2002) and (Schaal et al., 2003) provide an overview of proposed approaches for motion learning by imitation. As noted by Breazeal and Scasellati, the majority of algorithms discussed in the literature assume that the motions to be learned are segmented a-priori, and that the model training takes place off-line.

(Ogata et al., 2005) develop a connectionist architecture suitable for long term, incremental learning. In their work, a neural network is used to learn a navigation task during cooperative task execution with a human partner. The authors introduce a new training method for the recursive neural network, which avoids the problem of memory corruption. Their approach is based on neuroscience research, which has shown that the hippocampus is used as temporary (short term) memory, and that this memory is consolidated into long term memory during sleep in a process analogous to rehearsal. However, in this case, the robot learns only one task, and no hierarchical organization of knowledge takes place.

Hidden Markov Models have been a popular technique for human motion modeling, and have been used in a variety of applications, including skill transfer (Dillmann et al., 1999), sign language and gesture modeling and human motion modeling (Billard et al., 2006, Inamura et al., 2004). A common paradigm is *Programming by Demonstration* (PbD) (Dillmann et al., 1999). While PbD is a general paradigm, in many of the systems demonstrated thus far, the number of motions is specified a-priori, the motions are clustered manually and trained off-line, and then a static model is used during the recognition phase.

In order to learn from on-line demonstration, the system must group together similar motions on-line, without a-priori specifying the number of motions. While many off-line clustering approaches have been proposed in the literature (see (Jain et al., 1999) for a review), to the authors' knowledge, few algorithms consider on-line, incremental clustering of multi-dimensional time series data. One recent example is the work of (Rodrigues et al., 2008) describing the online divisive-agglomerative clustering approach for time series data. Time series data are compared via the Pearson's correlation coefficient, and an incremental procedure is developed to build a tree structure through on-line observation of the data.

This paper is based on the incremental clustering approach proposed by (Kulić et al., 2007a, Kulić et al., 2008b). An on-line, incremental learning algorithm is used to build an initial model of the motion space, as motions are perceived. In this approach, a Hidden Markov Model based representation is used to abstract motion patterns as they are perceived. Individual motion patterns are then clustered in an incremental fashion, based on intra model distances. The resulting clusters are then used to form a group model, which can be used for motion generation. Errors in the initial placement of both individual motions and motion groups are then also corrected on-line at a later time, in a process analogous to memory consolidation in humans.

1.2 Connections to Biological Models

A key question in biology and cognitive science is how humans and other primates acquire (learn) motion primitives. (Heyes and Ray, 2000, Heyes, 2001) propose the Associative Sequence Learning (ASL) theory of imitation. They postulate that learning is based on *action units*, which are the basic units of the majority of actions being observed. Learning proceeds via two sets of associative processes, resulting in horizontal and vertical links. The horizontal process forms sequence associations between sensory representations of the action units forming the demonstrated action. The vertical process forms associations between the sensory representation for each action unit and the associated motor representation of the same component. The ASL theory postulates that the development of the imitation mechanism is highly experience-dependent, consisting of correlation links between sensory and motor data which are formed over time. As also noted by (Calinon and Billard, 2007), our approach of modeling the sensory data flow as a set of state vectors linked temporally with a stochastic model (the HMM) and obtained incrementally over time, based on the experiences observed by the robot, conforms with the ASL theory. The HMM model learning corresponds to the horizontal process of the ASL model, which forms sequence associations between sensory representations of the action units.

A second area of study is the neural processes occurring during and immediately following learning. Neuroscience studies of motor memory formation have shown that following learning, the motor memory does not remain constant in the brain, but rather changes over time in a process termed memory consolidation (Stickgold, 2005, Krakauer and Shadmehr, 2006, Shadmehr and Holcomb, 1997, Diekelmann and Born, 2007). There appear to be two complementary memory consolidation processes: an initial stage of stabilization, through which the memories become resistant to interference, and a second stage, where system wide re-organization is performed (Stickgold, 2005, Diekelmann and Born, 2007). The first stage (the waking stage) occurs in the hours immediately following motor memory acquisition, while the subject is awake, and can be detected by evidence of disruptions corrupting the memory during a limited time window following initial acquisition (Krakauer and Shadmehr, 2006). The second stage (sleep-dependent stage) occurs during sleep, and can be detected by measuring performance improvements following sleep, without any further practice of the motion (Stickgold, 2005, Diekelmann and Born, 2007). During this type of consolidation, brain imaging studies show that brain regions active during memory formation are repeatedly reactivated during sleep as the motion representation is gradually re-distributed to different networks and brain regions, thought to signify a system-level consolidation (Diekelmann and Born, 2007).

(McClelland et al., 1995) describe the approaches to modeling learning and memory via connectionist systems. They highlight the importance of *interleaved learning*, whereby a particular item is not learned instantly, but is acquired gradually, through a series of presentations interleaved with exposure to other items.

In our approach, following observations of similar motions, an initial model of the abstracted motion is formed, corresponding to the waking stage. Then, at a later time, system wide re-organization is performed by selecting learned motions and reclassifying, therefore correcting initial clustering and hierarchy formation errors, in a process analogous to rehearsal memory consolidation found in the sleep-dependent stage.

2. Incremental Behavior Learning

In the proposed system, the task of the learning system is to autonomously extract and learn motion primitives from time series data obtained through on-line observation of a human demonstrator. A motion primitive is defined as full-body motion segment (i.e., an action unit (Heyes and Ray, 2000, Heyes, 2001)) which is re-used multiple times during task or behavior execution. A motion primitive may be described in terms of the joint or Cartesian coordinates. Motion primitives can be segmented autonomously from the continuous time-series data stream via stochastic segmentation (Kulić et al., 2008a).

Each time a new motion sequence is observed, the robot must decide if the observed motion is a known motion, or a new motion to be learned. In addition, over the lifetime of the robot, as the number of observed motions becomes large, the robot must have an effective way of storing the acquired knowledge for easy retrieval and organization. In the proposed approach, a hierarchical tree structure is incrementally formed representing the motions learned by the robot. Each node in the tree consists of a group model which represents a motion primitive, which can be used to recognize a similar motion, and also to generate the corresponding motion for the robot.

An overview of the algorithm and the incremental hierarchy formation is shown in Figure 1. The algorithm initially begins with one behavior group (the root node). Each time a motion is observed from the teacher, it is encoded into a Hidden Markov model (Figure 1(a)). The encoded motion is then compared to existing behavior groups via a tree search algorithm, using the symmetric model distance measure (Rabiner, 1989, Kulić et al., 2007b) (Figure 1(b)), and placed into the closest group (Figure 1(c)). Each time a group is modified, a hierarchical agglomerative clustering algorithm (Jain et al., 1999) is performed within the exemplars of the group (Figure 1(d)). If a cluster with sufficiently similar data is found, a child group is formed with this data subset (Figure 1(e,f)). The time series data of the motion examples forming the child group is then used to generate a single group model, which is subsequently used for both behavior recognition and generation. Therefore the algorithm incrementally learns and organizes the motion primitive space, based on the robot's lifetime observations. The algorithm pseudocode is shown in Figure 2. We assume that motion primitives can be well separated via a distance measure, such that clustering via a tree structure can be accomplished. Note that the tree structure is formed by creating a child node only when a similar grouping is found, and not by splitting nodes, thereby reducing the likelihood of overfitting.

```
procedure IncrementalCluster
    Step1 Encode observation sequence $O_i$ into an HMM $\lambda_i$
    Step2 Search the behavior tree for the closest group $\lambda_{Gj}$ to the current observation model $\lambda_i$, based on the inter-model distance
    Step3 Place $\lambda_i$ into the closest group $G_c$
    Step4 Perform clustering on all the exemplar motions within $G_c$
    Step5 If a sufficiently similar subgroup of motions is found, form a new group $G_n$, as a child of $G_c$, containing the observation sequences of the subgroup
    Step6 Using the observations sequences of the new subgroup, form the group model $\lambda_{Gn}$
end procedure
```

Figure 2: Segmenting Algorithm Pseudocode

Each newly acquired observation sequence is encoded into a Hidden Markov Model. Once the newly observed behavior is encoded, it is compared to existing groups (if any). Here, the distance between two models can be calculated (Rabiner, 1989) by Equation 1.

$$D(\lambda_1, \lambda_2) = \frac{1}{T}[logP(O^{(2)}|\lambda_1) - logP(O^{(2)}|\lambda_2)] \quad (1)$$

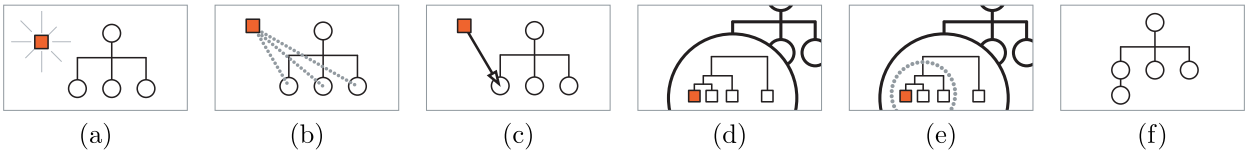

Figure 1: Overview of the Segmenting Algorithm (A square represents a data sequence, and a circle represents a group). (a) a new observation sequence is observed and encoded as an HMM; (b) the observation sequence is compared to existing groups via tree search; (c) the new sequence is placed in the closest existing group; (d) local clustering is performed on the modified group (zoomed in view of modified group); (e) a new subgroup is formed from similar motions in the modified group; (f) the subgroup is added to the tree as a child of the modified group.

where λ_1, λ_2 are two models, $O^{(2)}$ is an observation sequence *generated* by λ_2 and T is the length of the observation sequence. Since this measure is not symmetric, the average of the two intra distances is used to form a symmetric measure. This distance measure is based on the relative log likelihood that a generated sequence is generated by one model, as compared to a second model. It represents a Kullback-Leibler distance between the two models.

The repository of known groups is organized in a tree structure, so that the new observation sequence does not need to be compared to all known behaviors. The comparison procedure is implemented as a tree search. If the distance between the new observation and the cluster is larger than a threshold based on the maximum intra observation distance D^G_{max}, this cluster will not be considered as a possible match for the new observation sequence. If there are multiple candidate clusters, the new sequence is placed in the closest cluster. If there are no candidates, the new sequence is placed in the parent cluster. In the case of a new motion pattern which is completely dissimilar to any existing motion patterns, the motion pattern will be placed into the root node.

When a new observation sequence is added to a group, a clustering procedure is invoked on that group, to determine if a subgroup may be formed. The complete link hierarchical clustering algorithm is used to generate the hierarchical tree structure within a group (Jain et al., 1999). Clusters are formed based on two criteria: number of leaves in the subgroup, and the maximum proximity measure of the potential subgroup. The proximity measure is computed as follows:

$$D_{cutoff} = \mu - K_{cutoff}\sigma \qquad (2)$$

where D_{cutoff} is the distance cutoff value (i.e., only clusters where the maximum distance is less than this value will be formed), K_{cutoff} is a constant parameter, μ is the average distance between observations in the group, and σ is the standard deviation among all the distances in the node. If a new subgroup is generated, a new group model is trained using the raw observation sequences from all the group elements. The generated model is subsequently used by the robot to generate behaviors. The group model replaces the individual observations in the parent node.

If one of the group elements allocated to the new cluster is already a group model, the generated motion sequence based on that model is used for the training. In this case, a modified form of the re-estimation formulas for multiple observation sequences (Rabiner, 1989) is used. The algorithm is modified by over-weighting the group models, in order to account for the fact that there are multiple observation sequences stored in the generated model, and therefore more weight should be given to the group model, as compared to the individual observation sequences.

3. On-line Memory Consolidation

The incremental clustering algorithm described above has been shown to produce reliable results, robust against presentation order. In all the experiments performed thus far, no false positive errors have been reported at the leaf node (Kulić et al., 2008b). However, depending on the presentation order, two types of errors can occur: false negative errors (where the behavior is not classified at the correct hierarchy level), and tree structure errors (where the tree structure is not identical to the tree structure that would be observed during off-line clustering). These types of errors occur due to the incremental nature of the algorithm, where not enough information is available during early execution, when there are few examples, to find the correct segmentation boundary. However, as more data become available, these initial mistakes can also be corrected in an incremental, on-line fashion, analogous to memory consolidation in biological systems. The process is carried out by re-applying the incremental clustering procedure multiple times to data that is already known, at a later (possibly off-line) stage. This can be thought of as a type of rehearsal, similar to the idea proposed by (Ogata et al., 2005). Similar to (McClelland et al., 1995), we seek to improve performance through the benefits of interleaved learning.

Two types of corrections are proposed: corrections at the local level, dealing with misclassifications of individual exemplars, and corrections at the tree level, dealing with errors in structure formation. In the case of local correction, an individual exemplar model is selected, while in the case of structure correction, a group model is selected. For both types of corrections, the basic process is the same: a model is selected from the knowledge base, removed from its current location in the knowledge structure, and the incremental clustering algorithm used for classifying new motion, described in Section II, is re-applied. Various strategies can be considered for the selection of the next node to be analyzed for potential correction, for example, favoring recently classified models, or favoring those models where the group variability is large. In the experiments described below, a simple random selection strategy is employed. Figure 3 outlines the consolidation algorithm for exemplars, while Figure 4 shows the consolidation algorithm for the group models.

```
procedure MotionConsolidation
    Select sample node $n_s$
    Select sample motion $m_s$ from $n_s$
    $n_b$ = TreeSearch($m_s$) Search for the best match for the sample motion
    if $n_b = n_s$ then
        return (No correction required)
    else
        Remove $m_s$ from $n_s$
        if deletable($n_s$) then
            If few motions remain in $n_s$, and none are group motions
            Remove all $N$ motions from $n_s$
            Delete $n_s$ from tree
            for $i = 1 : N$ do
                Call $IncrementalCluster(m_i)$
            end for
        else
            Call $IncrementalCluster(m_s)$
        end if
    end if
end procedure
```

Figure 3: Memory Consolidation at the Exemplar Level

```
procedure NodeConsolidation
    Select sample node $n_s$, with parent node $n_p$
    Remove $n_s$ from $n_p$
    $n_b$ = TreeSearch($n_s$) Search for the best match for the group model of the node
    if $n_b = n_p$ then
        Return $n_s$ to $n_p$
        return (No correction required)
    else
        Add $n_s$ to $n_b$
    end if
end procedure
```

Figure 4: Memory Consolidation at the Node Level

Exemplar reconsolidation can also trigger changes in the tree structure, if a node is emptied of exemplars as a result of reconsolidations. In this case, the node is removed from the tree, and the remaining exemplars re-assigned by a call to the incremental clustering procedure.

4. Experiments

The proposed approach was tested on a data set containing a series of 9 different human movement observation sequences obtained through a motion capture system (Kadone and Nakamura, 2005). The data set contains joint angle data for a 20 degree of freedom humanoid model from multiple observations of walking (WA - 28 observations), cheering (CH - 15 observations), dancing (DA - 7 observations), kicking (KI - 19 observations), punching (PU - 14 observations), sumo leg raise motion (SL - 13 observations), squatting (SQ - 13 observations), throwing (TH - 13 observations) and bowing (BO - 15 observations).

In the experiments, the performance of the incremental clustering and hierarchy formation algorithm with and without memory consolidation is compared. Motion sequences are presented to the algorithm in random order. Motion sequences are presented one at a time, simulating on-line, sequential acquisition. After each motion is presented, the incremental clustering algorithm is executed, performing incremental clustering. When including memory consolidation, memory consolidation on one randomly selected node and one randomly selected motion is performed after each 10 new exemplars, and for 100 times at the end of the acquisition. In all of the tests performed, whether using reconsolidation or not, the algorithm correctly segments the behaviors such that the resulting leaf nodes represent the grouping that would be obtained with an off-line method. Out of 100 simulation runs performed, there was no cases of misclassification at the leaf nodes, showing that the final segmentation is robust to presentation order. Here, misclassification is defined as a false positive error (for example, a walk motion being misclassified as a punch motion). However, there were cases of false negative errors, where a motion which should have been recognized as a known motion was instead placed into a non-leaf node (for example, the root node).

Sample segmentation results for the non-consolidation algorithm are shown in Figs. 5, 6 and 7. Note that the actual order of node formation will vary depending on the motion presentation order. The average rate of false negative errors and the standard distribution of the false negative errors is shown for each motion in Tables 1 and 2. As noted before, no false positive errors were observed in any

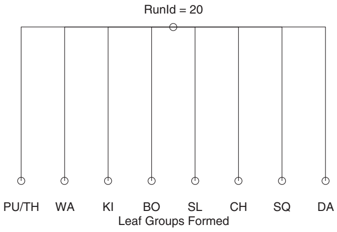

Figure 5: Sample Segmentation Result, $K_{cuttoff} = 1.2$

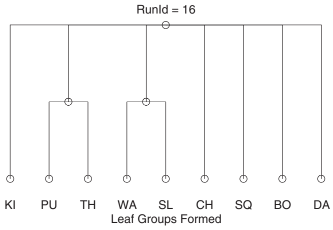

Figure 6: Sample Segmentation Result, $K_{cuttoff} = 0.9$

of the experiments performed.

The algorithm parameter K_{cutoff} (the parameter which controls when a new cluster is formed from an existing cluster) determines the resultant tree structure. A high value for K_{cutoff} (i.e., only clusters composed of a tight data set are formed) tends to result in a flat tree structure (as shown in Figure 5, while low values of K_{cutoff} result in a deep tree structure, as shown in Figure 6. As the cluster formation parameter is relaxed, deeper trees tend to be formed. However, the resulting tree structure tends to be dependant on the presentation order. In the case of a high cutoff value (see Figure 5), the resulting tree structure is flat, and fairly insensitive to presentation order. The resulting structure is consistent with off-line clustering result. In about 9% of cases, the 'dance' group fails to form, in contrast to the off-line clustering result, since this group contains the least examples. At the high cutoff value, the punch and throw motions are too similar to subcluster, resulting in a single hybrid generated motion (indicated as PU/TH in Figure 5). This is also indicated in the high false negative error rates for the punch and throw motions in Table 1, as the motions do not tend to be recognized as belonging to a distinct motion type, but are instead placed in the group node. The generated motion resulting from that subcluster is shown in Figure 8. As can be seen in the figure, the motion is an averaging of the two motions.

When low values of K_{cutoff} are used, nodes are quicker to form, and the resulting tree structure becomes more dependant on presentation order. The similarity level at which nodes will form is highly dependent on presentation order. Figures 6 and 7 show two examples of different tree structures formed, from two simulation runs. Note that the identified leaf nodes remain the same. In addition, using the lower cutoff value makes it easier to subdivide the similar throw and punch motions. This can be seen from the lower false negative rate in Table 2, as significantly more of the punch and throw motions are correctly recognized. Note that some punch throw motions still remain difficult to recognize, and are instead placed in the punch/throw parent node, which is classified as a false negative error in Table 2. Even though the cutoff level was the same for both experiments, the similarity level of the nodes formed differed, based on the presentation order. The result in Figure 6 is consistent with global clustering, while in the result shown in Figure 7, one node is incorrectly assigned. The CH node is incorrectly assigned to the PU/TH/KI/SQ branch of the tree, whereas global clustering would have assigned the CH node to the WA/SL branch. This type of error is due to the local nature of the algorithm, i.e. clustering is being performed when only a part of the data set is available. Therefore, there is a tradeoff when selecting the K_{cutoff} value between facilitating quick node formation and differentiation and introducing misclassifications in the hierarchy tree.

If the relationship between motions is not important for the task, a flat tree result is acceptable, and a high value of $K_{cuttoff}$ can be applied. However, it would be desirable to correctly extract both the leaf node groups, as well as a deeper tree, representing the hierarchy of motions. This tree information could then be used to analyze the relationships between motions and to accelerate learning of new motions belonging to the same branch of the tree. If

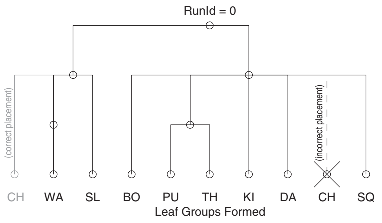

Figure 7: Sample Segmentation Result, $K_{cuttoff} = 0.9$

Table 1: False Negative Errors, $K_{cuttoff} = 1.2$. (An error rate of 1 indicates that all motions fail to be recognized)

Motion	Average	Std. Deviation
WA	0.0068	0.0316
CH	0.0273	0.1447
DA	0.0843	0.2725
KI	0.0621	0.0789
PU	0.9879	0.0891
SL	0.0254	0.1247
SQ	0.0377	0.1332
TH	0.9900	0.1000
BO	0.0047	0.0358

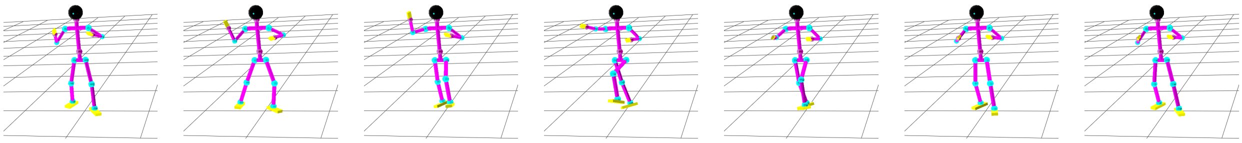

Figure 8: Generated Hybrid Punch/Throw Motion.

Table 2: False Negative Errors, $K_{cuttoff} = 0.9$. (An error rate of 1 indicates that all motions fail to be recognized)

Motion	Average	Std. Deviation
WA	0.0168	0.0786
CH	0.0407	0.0918
DA	0.1929	0.3934
KI	0.0526	0.0876
PU	0.4093	0.2240
SL	0.0492	0.1744
SQ	0.0546	0.1084
TH	0.5638	0.3585
BO	0.0360	0.0896

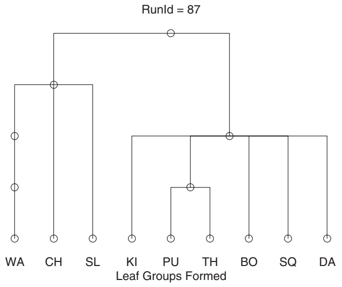

Figure 9: Sample Segmentation Result, $K_{cuttoff} = 0.9$, with memory consolidation

memory consolidation is applied, errors in the hierarchy tree can be corrected incrementally, allowing the algorithm to take advantage of fast node formation, while reducing both exemplar and node classification errors. Figure 9 shows a sample tree structure formed following incremental acquisition and memory consolidation. Table 3 shows the false negative error rate for each motion type when memory consolidation is applied. As can be seen from the table, false negative errors are significantly reduced when memory consolidation is used.

The resulting tree structures were also analyzed and compared to the global clustering result. Table 4 shows the results for the three cases considered: a high value of K_{cutoff} (1.2) with no memory consolidation, a low value of K_{cutoff} (0.9) with no memory consolidation, and a low value of K_{cutoff} (0.9) with memory consolidation. As can be seen from the results, memory consolidation reduces the mean tree error (computed based on edit distance), while producing deeper resultant trees.

Table 3: False Negative Errors, $K_{cuttoff} = 0.9$, Including memory consolidation. (An error rate of 1 indicates that all motions fail to be recognized)

Motion	Average	Std. Deviation
WA	0.0007	0.0050
CH	0.0007	0.0067
DA	0.0514	0.2192
KI	0.0326	0.0616
PU	0.1393	0.2397
SL	0.0100	0.0706
SQ	0.0300	0.1098
TH	0.2585	0.3639
BO	0.0027	0.0210

Table 4: Tree Analysis Results

K_{cutoff}	Consolidation	Mean Error	Mean Depth
1.2	No	2.11	2.08
0.9	No	1.96	2.94
0.9	Yes	1.21	3.81

5. Conclusions

This paper develops a novel approach towards on-line, long term incremental learning and hierarchical organization of whole body motion primitives. The learned motions are aggregates of the observed motions, which have been autonomously clustered during observation. The clustered motions are organized into a hierarchical tree structure, where nodes closer to the root represent broad motion descriptors, and leaf nodes represent more specific motion patterns. Errors made by the incremental clustering algorithm due to lack of sufficient data as a result of incremental acquisition are also corrected on-line via a mechanism for memory consolidation. The memory consolidation algorithm selects learned motions and re-classifies them using the current tree structure, in a procedure analogous to rehearsal. Both individual motion exemplars, as well as motion groupings can be processed in the same manner. In this way, the memory consolidation algorithm corrects both individual motion misclassifications and hierarchy formation errors, resulting in both improved classification and an improved structure over time.

Acknowledgment

This work is supported by the Japanese Society for the Promotion of Science grant 18.06754 and Category S Grant-in-Aid for Scientific Research 15100002.

References

Bentivegna, D. C., Atkeson, C. G., and Cheng, G. (2006). Learning similar tasks from observation and practice. In *IROS*, pages 2677–2683.

Billard, A., Calinon, S., and Guenter, F. (2006). Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54:370–384.

Breazeal, C. and Scassellati, B. (2002). Robots that imitate humans. *Trends in Cognitive Sciences*, 6(11):481–487.

Calinon, S. and Billard, A. (2007). Learning of gestures by imitation in a humanoid robot. In Nehaniv, C. L. and Dautenhahn, K., (Eds.), *Imitation and Social Learning in Robots, Humans and Animals*, pages 153–177. Cambridge University Press.

Diekelmann, S. and Born, J. (2007). One memory, two ways to consolidate? *Nature Neuroscience*, 10(9):1085–1086.

Dillmann, R., Rogalla, O., Ehrenmann, M., Zollner, R., and Bordegoni, M. (1999). Learning robot behaviour and skills based on human demonstration and advice: The machine learning paradigm. In *Int. Symp. on Robotics Research*, pages 229–238.

Heyes, C. (2001). Causes and consequences of imitation. *Trends in Cognitive Sciences*, 5(6):253–261.

Heyes, C. and Ray, E. (2000). What is the significance of imitation in animals? *Advances in the Study of Behavior*, 29:215–245.

Inamura, T., Toshima, I., Tanie, H., and Nakamura, Y. (2004). Embodied symbol emergence based on mimesis theory. *Int. Jnl. of Robotics Research*, 23(4–5):363–377.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323.

Kadone, H. and Nakamura, Y. (2005). Symbolic memory for humanoid robots using hierarchical bifurcations of attractors in nonmonotonic neural networks. In *IROS*, pages 2900–2905.

Krakauer, J. W. and Shadmehr, R. (2006). Consolidation of motor memory. *Trends in Neurosciences*, 29(1):58–64.

Kulić, D., Takano, W., and Nakamura, Y. (2007a). Incremental learning of full body motions via adaptive factorial hidden markov models. In *Int. Conf. on Epigenetic Robotics*, pages 69–76.

Kulić, D., Takano, W., and Nakamura, Y. (2007b). Representability of human motions by factorial hidden markov models. In *IROS*, pages 2388–2393.

Kulić, D., Takano, W., and Nakamura, Y. (2008a). Combining automated on-line segmentation and incremental clustering for whole body motions. In *ICRA*. 2591–2598.

Kulić, D., Takano, W., and Nakamura, Y. (2008b). Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *Int. Jnl. of Robotics Research*. To Appear.

McClelland, J. L., McNaughton, B. L., and O'Reilly, R. C. (1995). Why there are complimentary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457.

Nicolescu, M. N. and Matarić, M. J. (2005). Task learning through imitation and human-robot interaction. In Dautenhahn, K. and Nehaniv, C., (Eds.), *Imitation and social learning in robots, humans and animals: behavioral, social and communicative dimensions*. Cambridge University Press.

Ogata, T., Sugano, S., and Tani, J. (2005). Open-end human-robot interaction from the dynamical systems perspective: mutual adaptation and incremental learning. *Advanced Robotics*, 19:651–670.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286.

Rodrigues, P. P., Gama, J., and Pedroso, J. P. (2008). Hierarchical clustering of time series data streams. *IEEE Tran. on Knowledge and Data Engineering*. To Appear.

Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. *Phil. Tran. of the Royal Society of London B: Biological Sciences*, 358:537 – 547.

Shadmehr, R. and Holcomb, H. H. (1997). Neural correlates of motor memory consolidation. *Science*, 277:821–825.

Stickgold, R. (2005). Sleep-dependent memory consolidation. *Nature*, 437:1272–1278.