Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Control, Automation, Robotics and Vision (2010), pp: 453-458
DOI: 10.1109/ICARCV.2010.5707349

Hierarchical Learning Approach for One-shot Action Imitation in Humanoid Robots

Yan Wu and Yiannis Demiris
Department of Electrical and Electronic Engineering
Imperial College London
London SW7 2AZ, United Kingdom
{yan.wu08, y.demiris}@imperial.ac.uk

***Abstract*—We consider the issue of segmenting an action in the learning phase into a logical set of smaller primitives in order to construct a generative model for imitation learning using a hierarchical approach. Our proposed framework, addressing the "how-to" question in imitation, is based on a one-shot imitation learning algorithm. It incorporates segmentation of a demonstrated template into a series of subactions and takes a hierarchical approach to generate the task action by using a finite state machine in a generative way. Two sets of experiments have been conducted to evaluate the performance of the framework, both statistically and in practice, through playing a tic-tac-toe game. The experiments demonstrate that the proposed framework can effectively improve the performance of the one-shot learning algorithm and reduce the size of primitive space, without compromising the learning quality.**

***Index Terms*—imitation learning, one-shot learning, generative model, path planning, humanoid robots**

I. Introduction

In recent years, programming by demonstration (PbD) has received ever greater attention in Robotics research, especially in the domain of human-robot interaction (HRI). It provides a user-friendly teaching framework for robots (in particular, humanoids) to learn new skills from humans and other agents through imitation of actions. This has spawned a range of computational architectures that allow robots to match the demonstrated actions to its internal representations of equivalent motor commands [1]–[4].

Many of these models are capable of extracting and generalising important features for a given task, although they require numerous demonstrations of the same task to successfully learn the action. This means that in a complex environment, which involves non-expert users, any given PbD framework can help robots with high degrees of freedom (DoF) to drastically reduce the search space and hence speed up the learning process, but the issue of fatigue for giving demonstrations is still unresolved. Furthermore, the fact that most of such paradigms encode the perceptual information as a set of model statistics or internal states [5]–[8] hinders manual manipulation of the models by the demonstrators. This is of cardinal importance when a robot is placed in an unstructured environment for continuous learning because for every mistake in learning the model parameters, it takes many more perfect demonstrations to correct the model in these systems. In our previous work [9], we have proposed a One-Shot Imitation Learning Algorithm (OSILA) which stores the actions as human-readable and manipulable templates. This algorithm, addressing the "how-to" question in imitation, attempts to reduce the number of trials involved in learning and increase the model manipulability.

As suggested in [10] and [11], actions represented as movement primitives are a prerequisite for imitation learning with biological evidence in human. A significant proportion of PbD models focus on imitation learning at task level [8], [12], which are capable of performing the task with good accuracy. As compared to the primitive approach, actions learned with these models are more difficult to be useful in novel tasks. Even in many primitive-based imitation learning algorithms, the list of primitives is exhaustive and manually created with human intrinsic knowledge of basic actions. These primitives might not be the most natural set of basic actions and are limited for application in an unseen task. However, there is little literature in addressing the issue of breaking down a primitive into a logical set of smaller primitives at learning or the action phase.

In this paper, we propose a biologically-inspired Hierarchical Imitation Learning Approach that exploits the strength of the OSILA (HILA-OSILA) in a primitive-based learning framework. We present a system that handles both perception and action in robot learning, addressing the "how-to" question in imitation. In short, a clustering algorithm is applied to a given demonstration of action, breaking the demonstrated action down into a sequence of logical subactions with action tags. When a novel task, which can be expressed as a sequence of learned/stored templates and/or sub-templates in any order, is requested, the system makes use of a Finite State Machine (FSM) [13] and generates the task action by applying the OSILA to the sub-templates in the new constrained environment.

We showed in our previous work, by cross-validating the results on a set of 75 experiments conducted on human subjects [14], that the OSILA is capable of reproducing satisfactory path in imitating simple tasks. In the following sections of this paper, we will present the HILA-OSILA framework and extend the same dataset and evaluating metrics to test the statistical fitness for breaking-up the templates in contrast with the original mappings. We will also draw up discussions on an experiment to compare the implementation of this framework in performing a real-life tic-tac-toe game on a humanoid robot with our previous implementation detailed in [9].

II. The HILA-OSILA Framework

In this work, we consider the case of generating a new action, the *task*, in a 3-D environment which can be constructed by stitching portions of seen demonstrations, referred to as the *templates*, together. We assume that all required input features are observable from vision, i.e. in our case a pair of stereo cameras. This general framework of HILA-OSILA, shown in Fig. 1, is built upon three concepts:

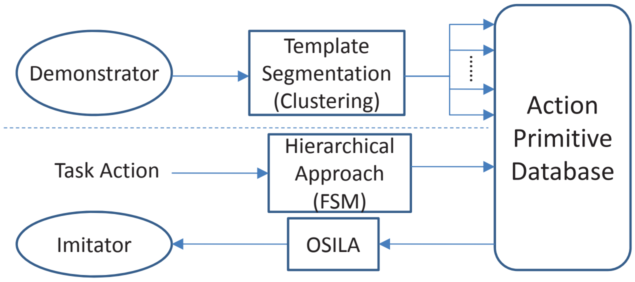

Demonstrator
Template Segmentation (Clustering)
Action Primitive Database
Task Action
Hierarchical Approach (FSM)
Imitator
OSILA

Fig. 1: **The schematics of the HILA-OSILA Learning Framework**. This framework consists of the two parts of imitation learning. The building block above the dotted-line denotes perception while that below the dotted-line denotes action. The three fundamental algorithms involved are denoted by the rectangles in the centre and explained below.

1) The basic building block of OSILA that acts as an efficient model for generating a given *task* based on constraints mapping with a matched *template*
2) A suitable template segmentation algorithm that segments a demonstrated *template* into a natural set of *sub-templates*
3) A hierarchical approach that generates an accurate action plan based on possible combinations of *template* and/or *sub-templates*

A. The Basic Building Block - OSILA

The OSILA is a template-based path imitation algorithm by invariant feature mapping. This one-shot learning algorithm consists of three components. Briefly, it first generates a spatial mapping of possible path locations from the *template* to the *task* based on minimal distortional energy warping between the corresponding spatial constraints in both scenarios. A probable route is then created from the time-series information associated with these possible locations using minimum-energy strategy. Finally, the generated route is subject to an *Interactive Plan Adjustment* strategy for route correction. Fig. 2 shows the schematics of OSILA and each of the component is briefly explained below.

1) Feature Distortion Warping: According to [15], if a set of feature points in space that maintain a fixed spatial relationship in two distinct scenes can be identified, a minimum distortion function can be used to describe the spatial correspondence of all points in both spaces using Thin Plate Spline (TPS) warping. Assuming a given scene-matching algorithm, such as SIFT [16], can provide a set of corresponding Cartesian coordinates from both the *template* and the *task* as the invariant control points(ICPs), it is possible to generate a set of candidate waypoints for the target trajectory in the *task* space using this feature distortion warping algorithm.

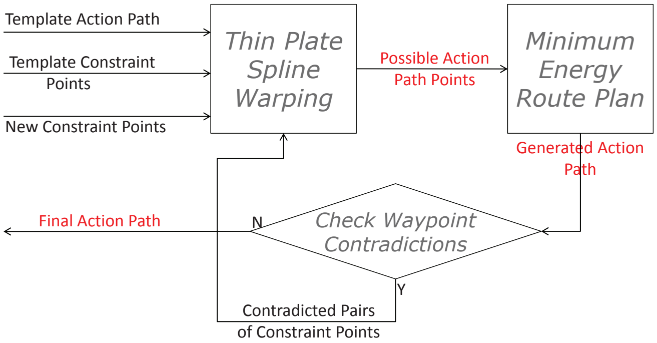

Template Action Path
Template Constraint Points
New Constraint Points
Thin Plate Spline Warping
Possible Action Path Points
Minimum Energy Route Plan
Generated Action Path
Final Action Path
N
Check Waypoint Contradictions
Y
Contradicted Pairs of Constraint Points

Fig. 2: **The schematics of OSILA**

2) Minimum-Energy Route Plan: Based on the time series encoded in the set of possible waypoints in the *task*, the goal-directed trajectory is simply represented by selecting the best point from each time stamp. As we believe that the main criterion for such route lies in energy consumption, we implement an efficient algorithm using dynamic programming [17] strategy to optimise for minimal translational energy.

3) Iterative Plan Adjustment: In a complex and variable environment, the trajectory generated using the above algorithm is then checked against extra invariant feature points present in the *task* space, such as forced waypoints or obstacles. If the route generated does not conform to these feature points, we use an iterative method [9] by finding the corresponding points in the *template* space and append these spatial constraints into the ICPs. The algorithm is then used again to generate a new route until the trajectory conforms to the constraints.

The ultimate aim of the OSILA is to produce a desirable path by imitation in a given scenario by inferring from a past demonstration. Furthermore, not only should the algorithm be able to generate such path with great level of stability, but more importantly have some resemblance to the path produced by human under similar circumstances. This has been demonstrated in our previous works [9], [14].

B. Template Segmentation

As we believe that an action constitutes a sequence of basic movements [18], PbD should therefore have the robustness to learn, segment and reproduce a given action as a chain-event of subactions. This is useful even from an engineering point of view:

1) It significantly reduces the redundancy involved in learning. For instance, a robot is shown how to move an arm to a range of objects, grasp them and bring them back to the demonstrator in turn. It is subsequently required to imitate all these actions. We can see that the redundancy is at actuating the arm to the object and back. Therefore, if the robot smartly learns these actions as a series of basic movements, the redundancy can be minimised.
2) The learned subactions can be integrated in a generative model to produce new actions without having to learn the task. For example, a demonstrated action consists

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Control, Automation, Robotics and Vision (2010), pp: 453-458
DOI: 10.1109/ICARCV.2010.5707349

of a sequence of Subactions A-B-C. The robot is then asked to perform an unseen action that constitutes the sequence of C-B-A-A-C. With the proposed generative model, the robot no longer has to learn this new action.

3) In contrast to the traditional primitive-based imitation learning, decomposing and learning the basic primitives at perception phase gives more generative flexibility in adopting new primitives/subactions without subjecting and limiting these subactions to human intrinsic knowledge of basic actions.

To simplify the problem, we shall assume that the task actions consist of simple subactions and are separable in spatial-temporal manner. Thus, in these kinds of situations, we can apply the k-means clustering algorithm [19] for template segmentation. In more complex situations, we believe that a more robust algorithm such as the one proposed in [20] is capable of detecting changes in parameter space and time.

C. The Hierarchical Approach

In [21], the authors suggest that there exists a hierarchical control structure in the human cortex that executes actions in terms of superordinate chunks, simple chunks and single acts. Since an action can be broken down as a series of subactions, we can tag the action and each of the subactions in such a hierarchy. Multiple action and subactions can be integrated into a new action in a hierarchical manner. Fig. 3 is an illustration of this approach. Assuming a robot learns compound actions H and I as well as subactions D and F with a template segmentation algorithm, it basically has learned all subactions from A to G. When the robot is requested to perform an unseen action J, it breaks the action down hierarchically into a minimum set of subactions H-D-E-I. This can be modelled and implemented as an FSM.

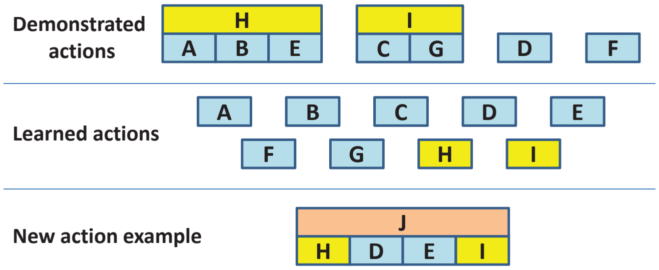
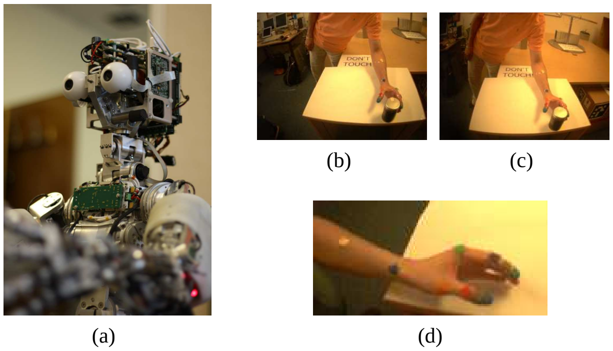


Fig. 3: **An example of the Hierarchical Approach of Template Integration**

III. Experiments

We implement and validate the HILA-OSILA framework with two sets of experiments on the iCub (Fig. 4a), a humanoid robot developed by the RobotCub Consortium[1]. In Experiment A, we evaluate the relative merit of segmenting the templates against that of the unsegmented ones in terms of statistical fitness. In Experiment B, a practical application of the framework is deployed to perform the same task on the actual robot with OSILA and HILA-OSILA. Throughout the experiments, demonstrations were captured by the stereo cameras on the iCub with camera frame rate set at 20Hz and frame resolution set at 320 X 240 pixels (example shown in Fig. 4b & 4c). Markers were placed on demonstrators (e.g. Fig. 4d) to simplify the task of tracking the points of interest.

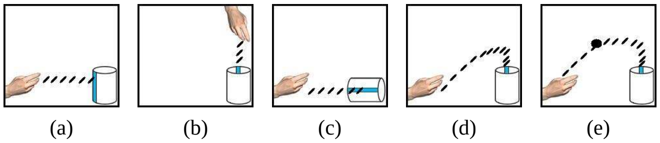


(a) (b) (c) (d)

Fig. 4: **The experiment set-up for testing the path planning algorithm.** The iCub in (a) is developed by the RobotCub Consortium. It has a total of 53 DoFs, while each of its arms has 16. (b) and (c) are an instance of a human subject with markers captured by the left and right cameras of the iCub respectively.(d) shows the locations of markers placed on the left arm of the human subjects in Experiment A.

A. Experiment A

In this experiment, we test the hypothesis on the improvement of performance accuracy and robustness in path generation using our HILA approach. The benchmarking dataset of 75 trials on 5 different experiments was taken from [14] and illustrated in Fig. 5. Each of the experiments consists of 15 observed paths.



(a) (b) (c) (d) (e)

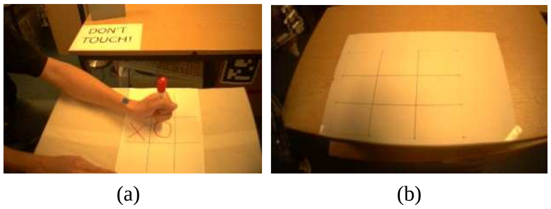
Fig. 5: **The sketches of the 5 conducted Experiments.** The hand positions in the diagrams indicate the starting points of the experiments. The subjects have also been requested to approach the object with their fore-arms orthogonal to the blue strips indicated in the diagrams. All experiments are constrained as planar movements apart from (c) where the demonstrators have to approach the object from the top. The black patch indicated in (e) denotes the waypoint area the subject have to navigate their arms through. The hypothesised paths are denoted by black slashes in the diagrams. Detailed descriptions of the experiments can be found in [14].

The spatial constraints of each path are mapped to the constraints of all other paths, including its own, so that a set of 75×75 paths can be generated by warping the input path based on the constraints mapping according to the OSILA model. To test the stated hypothesis, we simulate the template segmentation, making it a controlled variable, by dividing the path into $N \in 2 \ldots 4$ equal segments and input into the HILA-OSILA model to generate three other sets of corresponding paths. The set of 4 generated paths are then statistically compared to the path demonstrated by human under the same spatial constraints.

[1] www.RobotCub.org

B. *Experiment B*

The task of this experiment is to investigate the performance of the HILA-OSILA framework on the iCub by playing the tic-tac-toe game. During the perception phase, shown in Fig. 6a, the iCub is given one single demonstration of how to move to a grid space, place a mark and move away. While the human subject is instructed to demonstrate a planar movement, the iCub is to play the game in a new grid space of different size at a completely new location with its arm initially parked at a random location above the grid, shown in Fig. 6b. In this experiment, we assumed that the pen was always on the hand of the iCub and the invariant features were the four corners of the cell on the grid and the starting position of the arms. In contrast with our previous experiment on the same game [9], we aim to remove any redundancy in terms of unnecessary subactions. We first segment the demonstrated template using k-means clustering, then according to instruction build an ad-hoc deterministic FSM to execute the action.



(a) (b)

Fig. 6: **Experimental setup for Experiment B.** (a) shows how the demonstrator drew a circle in a A4-sized grid. The iCub was expected to play the game in scene (b), where the grid was 20% smaller, rotated and 20cm above that in (a).

C. *Implementation of the Algorithm*

From Fig. 4b & 4c, we can see that both cameras on the iCub have some degree of fish-eye distortion. Thus, before processing the captured frames, we undistort the images with a set of calibration parameters discussed in [22]. The marker positions are extracted using the technique presented in [23]. As we know that there is uncertainty in the image, assuming the extraction process treats the noise properly, the least amount of uncertainty associated is therefore 0.5 pixel. We, thus, applied Gaussian Radial Basis Function (RBF) smoothing with smoothing parameter of 0.5 pixels to the extracted paths. In the OSILA module, we also relax the threshold of the distortion to 2 pixels from the mapped locations. The final 3-D iCub-body-centred action path can be found by using independently generated paths from both cameras with their the intrinsic parameter. In Experiment B, the intended path is then passed to the inverse kinematics module of the iCub for execution.

D. *Statistical Performance Evaluation*

To evaluate the statistical fitness of the model, we introduce two performance metrics, namely Correlation Coefficient and Mean Squared Difference. We denote the generated path as $\mathbf{m}$ and that to be compared as $\mathbf{m}'$. Both consist of N corresponding waypoints.

1) Correlation Coefficient (R^2): Assuming that the proposed algorithm is an estimation of the resulted path generated by human under similar circumstances, R^2 is an indicator of how likely our proposed algorithm can be used to predict paths produced by human.

$$R^2 = \frac{\sum_{i=1}^{N}(\mathbf{m}_i - \bar{\mathbf{m}}) \cdot (\mathbf{m}'_i - \bar{\mathbf{m}}')}{\sqrt{(\sum_{i=1}^{N}(\mathbf{m}_i - \bar{\mathbf{m}})^2)(\sum_{i=1}^{N}(\mathbf{m}'_i - \bar{\mathbf{m}}')^2)}} \qquad (1)$$

where $\bar{\mathbf{m}}$ denotes the arithmetic mean of $\mathbf{m}_i$.

2) Mean Squared Difference (MSD): We make use of MSD to estimate the squared difference between the calculated path and the demonstrated path to gauge how close the generated paths are to human demonstrated ones.

$$MSD = \frac{1}{N}\sum_{i=1}^{N} \parallel \mathbf{m}'_i - \mathbf{m}_i \parallel^2 \qquad (2)$$

In practice, we cannot ensure all demonstrations are completed at the same duration. Since both performance metrics require the input vectors to be of the same lengths, we shall employ the cubic spline interpolation to lengthen the path with fewer waypoints to match that of the longer one.

IV. Results and Discussions

A. *Experiment A*

As we were to generate four paths per input-output pair of spatial constraints in the algorithm and there were 75 trials in total, this produced a 75×75×4 tensor of paths for cross-validation. We assessed the performance of the framework primarily in two ways - stability and generalisation.

Stability of the algorithm is achieved when the output path resembles the input one if the same spatial points are used as both input and output constraints. Assuming a given demonstration is the optimal path, any self-mapping case should preserve maximally the input path. TABLE I shows the correlation coefficients for the 75 self-mapping cases sorted according to number of segments. We can see that in all cases, the confidence indicator is almost 100% which suggests that with and without the use of subaction templates, the framework maintains output stability as the spatial constraints are fixed in these cases.

TABLE I: The averaged Correlation Coefficient of self-mapping cases grouped in experiments.

No of Segments	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
1	0.996	0.999	0.991	0.991	0.994
2	0.999	1.000	0.999	1.000	1.000
3	0.999	1.000	0.999	1.000	1.000
4	0.999	1.000	0.999	1.000	1.000

By pulling cases of similar input-output constraints together, i.e. grouping by experiments, if the statistics suggest that the output paths generated match well with the intended paths, the algorithm is said to generalise well. TABLEs III and II tabulate the statistics grouped by input-output experiment pairs. As shown in the first line of each row in TABLE II, when only

TABLE II: The averaged Correlation Coefficients for mapping from one experiment to another. Columns indicate input while rows indicate output. Within each cell, the statistics correspond to No of Segments from 1 to 4 respectively.

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Exp 1	0.961	0.953	0.817	0.414	0.380
	0.979	0.941	0.860	0.800	0.791
	0.987	0.976	0.948	0.949	0.934
	0.989	0.984	0.977	0.978	0.966
Exp 1	0.993	0.995	0.941	0.740	0.711
	0.970	0.998	0.877	0.972	0.964
	0.989	0.999	0.970	0.994	0.990
	0.995	0.999	0.988	0.997	0.995
Exp 3	0.861	0.444	0.891	0.774	0.747
	0.910	0.896	0.941	0.877	0.847
	0.952	0.948	0.967	0.963	0.954
	0.966	0.968	0.982	0.980	0.973
Exp 4	0.757	0.817	0.871	0.957	0.938
	0.872	0.845	0.798	0.977	0.967
	0.929	0.922	0.938	0.987	0.981
	0.944	0.954	0.969	0.992	0.987
Exp 5	0.835	0.816	0.872	0.885	0.962
	0.875	0.879	0.840	0.966	0.977
	0.937	0.935	0.935	0.979	0.987
	0.952	0.960	0.965	0.984	0.991

TABLE III: The averaged Mean Squared Difference for mapping from one experiment to another. Columns indicate input while rows indicate output.Within each cell, the statistics correspond to No of Segments from 1 to 4 respectively.

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Exp 1	176	298	1222	7388	8390
	74	177	682	538	562
	31	63	133	120	156
	18	30	58	55	76
Exp 2	117	57	1899	2809	5095
	436	26	2448	378	444
	152	8	356	78	136
	70	4	122	28	56
Exp 3	799	2832	399	962	2869
	385	316	231	619	852
	143	117	122	110	140
	70	55	62	57	68
Exp 4	1548	667	1164	203	343
	606	551	2317	148	176
	211	219	397	69	94
	115	111	175	41	59
Exp 5	1139	834	1286	459	280
	758	478	1963	222	133
	255	261	428	124	75
	151	172	215	92	44

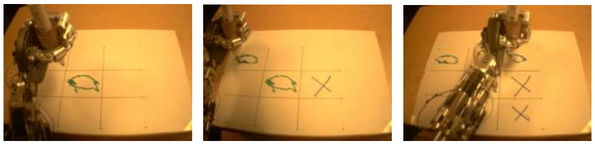
OSILA is in use (no segmentation of the path), 32% of the results have $R^2 \leq 0.8$ which suggests that the algorithm cannot generalise well in these experiment pairs, such as mapping Exp 5 path into Exp 1. This is also confirmed by statistics from TABLE III.

In previous work [14], we suggested that this was due to the lack of complete invariant information in the complex cases. Thus, by breaking the paths into smaller segments, we introduced extra spatial constraints into mapping. This should improve the performance of imitation. From TABLE II, we can see that as soon as template segmentation is introduced, 92% of the correlation coefficient are greater than 0.8. When the number of segment goes up to 4, nearly all R^2 are greater than 0.95. We thus believe that our previous claim can be sustained and the HILA-OSILA algorithm can reduce the cost and burden of repeated demonstrations, while maintaining good generalisation.

B. *Experiment B*

Fig. 7 captured a series of snapshots of the game played by iCub. In our previous work [9], the iCub imitated the demonstrated action at *task* level, i.e. after placing a mark, the arm was moved back to a parking position (Fig. 8b). Under the HILA-OSILA framework, the template segmentation module (with $k = 3$ for k-means clustering) separated the demonstrated action into 3 logical subactions (Fig. 8a), which could be tagged as reach cell, draw circle in cell and retract. The iCub was then instructed to draw circles in the cells with exact same sequence from previous experiment. However, after each drawing, the FSM had the next state as reaching cell instead of retracting until the end. The new path is plotted in Fig. 8c.

From Fig. 7, we can see that the size and the position of marks were fairly accurately drawn by iCub. Although limited by the inverse kinematics module, the shape drawn did not affect the discrimination between the 2 different symbols in the game. Close comparison of the circles generated in Fig. 2 and in Fig. 3 suggests that breaking up the templates into subactions did not undermine generalisation. The correlation coefficient of the corresponding circles is greater than 0.9. In fact, the circles imitated at the same location (centre) by both algorithms are statistically congruent to the demonstrated one.



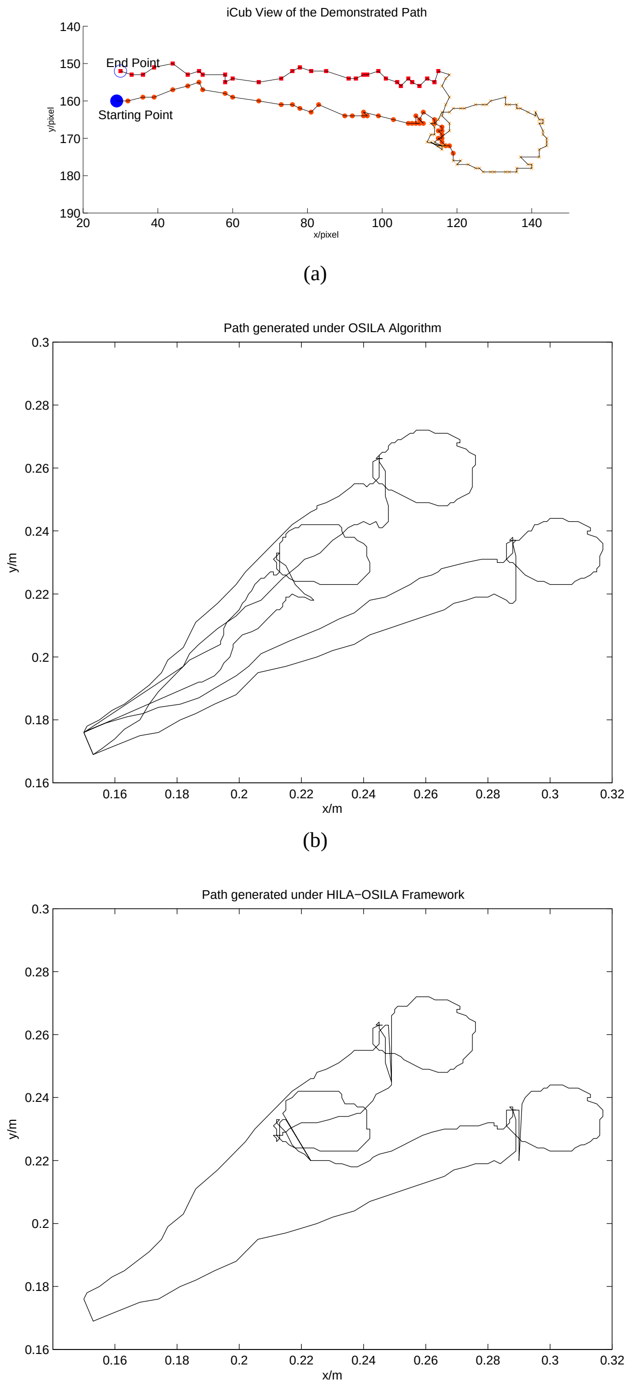
(a) (b) (c)

Fig. 7: **iCub playing tic-tac-toe.** The iCub started the game by marking the centre cell (a) with the path learned from demonstration. Subsequently, it then marked on top-left (b) and top-right (c) cells

V. Conclusion

In this paper, we have presented a biologically-inspired one-shot hierarchical primitive-based learning framework for robot path imitation. This algorithm has been implemented and statistically evaluated using cross-validation results from the paths demonstrated by human subjects with a range of template segmentation sizes. It has also been implemented to allow a humanoid robot to play the tic-tac-toe game by reducing redundant movements in execution. The experimental results show that the HILA-OSILA framework is capable of reproducing highly satisfactory paths by imitating simple tasks as compared to the OSILA algorithm alone. However, the experiments have been conducted with assumptions, such as sufficient invariant feature points were given for mapping and untested in a dynamic environment. This inexpensive algorithm is capable of not only pure imitation, but also with generative component to increase primitive skill-sets. We plan to extend our research to include automatic detection of known subactions in the primitive database and segment these

Preprint version; final version available at http://ieeexplore.ieee.org
IEEE International Conference on Control, Automation, Robotics and Vision (2010), pp: 453-458
DOI: 10.1109/ICARCV.2010.5707349



(a)

(b)

(c)

Fig. 8: **Paths imitated from one single demonstration.** (a) shows the demonstrated path seen from the left camera of the iCub segmented by k-mean clustering algorithm with $k = 3$. (b) shows the generated paths during the game by OSILA, while (c) shows the HILA-OSILA generated paths.

templates online in the perception phase possibly by using the HAMMER architecture proposed in [11].

Acknowledgment

The authors gratefully acknowledge the invaluable comments and continued support from members of the BioART Lab. The authors would also like to express their gratitude towards the participants in the experiments.

References

[1] A. Alissandrakis, C. Nehaniv, and K. Dautenhahn, "Imitation with ALICE: Learning to imitate corresponding actions across dissimilar embodiments," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 32, no. 4, pp. 482–496, 2002.

[2] A. Billard, "Learning motor skills by imitation: a biologically inspired robotic model," *Cybernetics and Systems*, vol. 32, pp. 155–193, 2001.

[3] Y. Demiris and G. Simmons, "Perceiving the unusual: Temporal properties of hierarchical motor representations for action perception," *Neural Networks*, vol. 19, no. 3, pp. 272–284, 2006.

[4] J. Peters, S. Vijayakumar, and S. Schaal, "Reinforcement learning for humanoid robotics," in *Proceedings of the Third IEEE-RAS International Conference on Humanoid Robots*, 2003, pp. 1–20.

[5] J. Demiris and G. Hayes, "Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model," *Imitation in animals and artifacts*, pp. 327–361, 2002.

[6] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109–116, 2004.

[7] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains," *The International Journal of Robotics Research*, vol. 27, no. 7, pp. 761–784, 2008.

[8] S. Calinon, F. Guenter, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," in *Proceedings of 2006 IEEE International Conference on Robotics and Automation. ICRA 2006.*, 2006, pp. 2978–2983.

[9] Y. Wu and Y. Demiris, "Towards One Shot Learning by Imitation for Humanoid Robots," in *Proceedings of 2010 IEEE International Conference on Robotics and Automation. ICRA 2010.*, May 2010, pp. 2889–2894.

[10] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[11] Y. Demiris and B. Khadhouri, "Hierarchical attentive multiple models for execution and recognition of actions," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 361–369, May 2006.

[12] D. Grimes, R. Chalodhorn, and R. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," *Proceedings of Robotics: Science and Systems. RSS 2006.*, pp. 199–206, 2006.

[13] T. Chow, "Testing software design modeled by finite-state machines," *IEEE Transactions on Software Engineering*, pp. 178–187, 1978.

[14] Y. Wu and Y. Demiris, "Efficient template-based path imitation by invariant feature mapping," in *Proceedings of 2009 IEEE International Conference on Robotics and Biomimetics. ROBIO 2009.*, Dec 2009, pp. 913–918.

[15] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.

[16] D. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[17] R. Bellman, "Some problems in the theory of dynamic programming," *Econometrica: Journal of the Econometric Society*, pp. 37–48, 1954.

[18] J. Allen, "Towards a general theory of action and time," *Artificial intelligence*, vol. 23, no. 2, pp. 123–154, 1984.

[19] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1967, pp. 281–297.

[20] B. Takács and Y. Demiris, "Balancing Spectral Clustering for Segmenting Spatio-temporal Observations of Multi-agent Systems," *IEEE International Conference on Data Mining*, pp. 580–587, 2008.

[21] E. Koechlin and T. Jubault, "Broca's area and the hierarchical organization of human behavior," *Neuron*, vol. 50, no. 6, pp. 963–974, 2006.

[22] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, p. 1106, 1997.

[23] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactiverobots," in *Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2000.)*, vol. 3, 2000, pp. 2061–2066.