Proceedings of the 2009 IEEE
International Conference on Robotics and Biomimetics
December 19 -23, 2009, Guilin, China

Efficient Template-based Path Imitation by Invariant Feature Mapping

Yan Wu, and Yiannis Demiris, *Senior Member*, IEEE

***Abstract*— We propose a novel approach for robot movement imitation that is suitable for robotic arm movement in tasks such as reaching and grasping. This algorithm selects a previously observed path demonstrated by an agent and generates a path in a novel situation based on pairwise mapping of invariant feature locations present in both the demonstrated and the new scenes using minimum distortion and minimum energy strategies. This One-Shot Learning algorithm is capable of not only mapping simple point-to-point paths but also adapting to more complex tasks such as involvement of forced waypoints. As compared to traditional methodologies, our work does not require extensive training for generalisation as well as expensive run-time computation for accuracy. Cross-validation statistics of grasping experiments show great similarity between the paths produced by human subjects and the proposed algorithm.**

***Index Terms* - movement imitation, path planning, grasping, learning by imitation**

I. INTRODUCTION

Robot movement imitation is key to robotic programming by demonstration (RbD). It addresses the ability of a robot to perceive, learn, generalise and reproduce the physical action from a demonstration for a given task with extrinsic parameters [1]. An increased interest in learning algorithms that will equip robots to learn by imitation of actions from both humans and other robotic agents has resulted in many research directions within the area. To address issues concerning attention and perception, a hierarchical model was presented in [2]. For methodologies to teach robots to learn new skills, while Muench et al proposed to ask the demonstrators explicit questions in order to generalise an action[3], some others made use of reinforcement learning to extract useful information ([4], [5]). Many of these research directions can be encapsulated by the 4 "W"s & 1 "H" of imitation - namely who, where, when, what and how.

In this paper, we propose an algorithm to address the "how-to" question in imitation. We introduce a novel computational model for path planning by imitation which makes use of pattern matching - aligning feature points in the demonstration to the actual situation - to generate a motion path for the new scenario. The following sections of this paper will present some related work to our approach followed by the detailed decsription of the methodology. We will also discuss the results taken from the cross-validation of 75 experimental trials conducted on human subjects.

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

II. RELATED WORK

To reproduce a trajectory in an unseen situation, simple copying of an observed path does not work well. For instance, rolling a ball into a catchment area straight ahead cannot be directly applied to rolling it with a subtended angle. Moreover, in the new situation, there might be additional constraints, such as forced waypoints, that require some re-mapping along an imitated route. Thus, a robot must have the ability not only to generalise, but more importantly to adapt. Research works in the domain have focused on approaches either to find a unique yet exact corresponding imitation of the previously demonstrated trajectories ([6], [7]), or to generalise a new path based on a subset of competent tasks so as to accommodate additional constraints[8].

In order for most imitation algorithms to work successfully, several demonstrations of a single task have to be performed ([8], [9]). This is a time-consuming and tedious process for human subjects. Thus, One-Shot Learning, a popular area in machine learning, can be very beneficial in speeding-up learning process and reducing fatigue in giving demonstrations. There have been very limited research to address this issue. Most of such works focus on explanation-based methods[10] which are slightly difficult to be put into actual practice.

Most of the state-of-the-art paradigms in path imitation generalise observed behaviours into a set of intrinsic model parameters, for example, in algorithms employing the Hidden Markov Models[11], the generalised parameters are the internal states with associated probabilities. In [12], Friedrich et al argue that such paradigms limit the ability of user interaction after demonstration. As the abstract meaning of these parameters is hidden to or too complex for human, these algorithms deter direct user interaction/intervention. For example, when a demonstrator accidentally demonstrates a wrong movement, the probable solution is either to redo all demonstrations or to perform many more correct demonstrations to rectify the parameters.

In the field of both image processing and motion plan adaptation, research works have been done to morph a scene in the current context into a new scene by minimising the energy involved in morphing([13], [14]). To address the "how-to" question, we can perceive a path imitation problem as plan adaptation whereby the original path(in our case, from demonstration) with the set of additional environmental features is projected to a warping space. We then generate an imitated path based on the preservation of the spatial relationship between the imitated and the original features in that space.

III. PROBLEM FORMULATION

In this work, we consider the general case of path planning in a 3-D environment inferred from the demonstration of a similar task. To address this problem, we assume that all the required input features are observable from vision, i.e. in our case a pair of stereo cameras. Path planning for a planar task can be therefore considered as a special case of this work which requires only one camera.

The ultimate aim of the algorithm is to produce a desirable path for a given scenario. Generation of the path should be an inference from a past demonstrated case. 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.

For a given demonstration viewed from each of the pair of stereo cameras, we describe the motion path as a set of p discretised spatial feature points $\mathbf{m}_l : (x_l, y_l), l \in \{1...p\}$ in the time series images. We also assume that the target destination in the scene can be fully described by a set, F of n point-like features, where each is described by $(\mathbf{a}_i, \mathbf{A}_i)$, $i \in \{1...n\}$. While $\mathbf{a}_i$ represents the Cartesian coordinates of the feature point, $\mathbf{A}_i$ encapsulates additional information that will help to match invariant points of similar objects/targets, such as SIFT[15] features and textual features. We refer to such a given demonstration as a *template* and a new situation as the *task*.

In a new situation described similarly by a set F' : $(\mathbf{a}_j, \mathbf{A}_j), j \in \{1...n'\}$, we assume that there exists a confidence function for correspondence, $f_c(\mathbf{A}_i, \mathbf{A}_j)$, where

$$f_c(\mathbf{A}_i, \mathbf{A}_j) = \begin{cases} 0 & \text{if } \mathbf{A}_i \ \& \ \mathbf{A}_j \text{ are uncorrelated} \\ 1 & \text{if } \mathbf{A}_i \text{ matches } \mathbf{A}_j \end{cases} \quad (1)$$

Depending on the context, we can employ different feature mapping algorithms to match the **A**s. SIFT algorithm is one common example. We, thus, can identify a maximum k pairs of coordinates in the image space of both the *task*, $\mathbf{a}_j$, and one of the learned *templates*, $\mathbf{a}_i$, where $f_c(\mathbf{A}_i, \mathbf{A}_j) = 1$, $k \leq n$, $k \leq n'$. This k pairs of coordinates should also include the pair of starting positions in the *task* and the *template*.

In a general path planning situation, we might be given more features in both the *task* and the *templates*, such as objects at far sight and textual features of the background. However, there are cases which inclusion of such features generates excessive output distortion which is undesirable. Thus, we should not impose the matching constraints for such features in order to preserve the spatial relationship between the cardinal features and generate a route for the *task*, $\mathbf{m}' : (x, y)$.

In a more complex situation, when the agent is required to pass through waypoints or to avoid enroute obstacles, these extra feature points can be also described by an additional set $F_x(\mathbf{b}_i, \mathbf{B}_i) \subseteq F$. If F_x exists in a *task*, these additional feature points should be considered only when the planned path $\mathbf{m}'$ contradicts $\mathbf{b}_i$.

IV. METHODOLOGY

In this section, we will describe, in details, our novel approach for path imitation. Briefly, we first generate a distortional mapping of each spatial point present in the *template* path into a set of possible locations based on minimal distortional energy between the k pairs of coordinates extracted from feature sets F and F' as well as the pair of starting positions. Based on the time series information of the *template* path, the *task* path is created from the cloud of possible waypoints using minimum-energy strategy.

A. Feature Distortion Warping

We define the k cartesian coordinates $\mathbf{a_i}$ from the *template* as the invariant control points(ICP) **Pi** and the corresponding ICPs $\mathbf{a_j}$ in the *task* **Po**. If we can define the mapping from $\mathbf{Pi_w}$ to $\mathbf{Po_w}$ as a function f, to minimise the distortion of feature in space is equivalent to minimise the following energy function [16]:

$$E = \sum_{w=1}^{k} \parallel \mathbf{Po}_w - \mathbf{Pi}_w \parallel + \lambda E_f \quad (2)$$

where

$$E_f = \int \int_{R^2} (f''_{xx} + 2f'_{xy} + f''_{yy}) dxdy \quad (3)$$

The introduction of the *regularisation parameter*, λ, in (2) is to trade-off between the exact matching of points and the smoothness, which is particularly useful in the presence of noise. According to [16], the mapping function f shown in (2) is defined as

$$f(x,y) = a_0 + a_x x + a_y y + \sum_{i=1}^{k} \omega_i \phi(\parallel (x_i, y_i) - (x, y) \parallel) \quad (4)$$

where

$$\phi(r) = r^2 log(r) \quad (5)$$

(5) is a 2^{nd} order polyharmonic spline commonly known as a Thin Plate Spline. In order to ensure that E_f exists, the 2^{nd} derivatives of $f(x,y)$ must be square integrable, i.e. the following three conditions have to be met:

$$\sum_{i=1}^{k} \omega_i = 0 \quad (6)$$

$$\sum_{i=1}^{k} \omega_i x_i = \sum_{i=1}^{k} \omega_i y_i = 0 \quad (7)$$

By letting $\Phi_{ij} = \phi(\parallel (x_i, y_i) - (x_j, y_j) \parallel)$ and $\upsilon_i = f(x_i, y_i)$, based on (4 - 7), we can form a linear equation as follows:

$$\begin{bmatrix} \Phi & L \\ L^T & O \end{bmatrix} \begin{bmatrix} \omega \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \upsilon \\ \mathbf{0} \end{bmatrix} \quad (8)$$

where ω is a column vector of ω_i, $\mathbf{a} = [\ a_0 \quad a_x \quad a_y\ ]^T$ and the i^{th} row of L, $L_i = [\ 1 \quad x_i \quad y_i\ ]$.

In [17], it has been shown that the square matrix in (8) is non-singular. Thus, we can define the upper left $k \times k$ submatrix of the inverse of this square matrix by M'_k. It can be shown that $E_f \propto \upsilon^T M_k' \upsilon = \omega^T M \omega$. Thus, the optimal solution of ω and a with minimum bending energy can be solved either by analytical method or approximation methods described in [18] and [19] depending on the importance of accuracy or computational cost.

For each $\mathbf{m}_l$ in the given set of p spatial coordinates in the *template*, there exists a number of mapped coordinates $\mathbf{m}'_{lq} : q \geq 0$ in the *task* defined by (4). As these coordinates are often at sub-pixel level, the minimum energy enforcement is relaxed and thresholded to accommodate neighbouring coordinates.

B. Minimum-Energy Route Plan

Given the time series point clouds of $\mathbf{m_l'}$, the goal-directed movement is simply represented by stepping through the variable l, and connect the best points from each $\mathbf{m}'_{lq}$ to form the trajectory. We shall make use of the translational energy as the cost function to derive the *task* path. The path through these best points should minimise such cost function. Thus, the optimisation criterion C_{eng} is to minimise this cost function which is proportional to the sum of changes in positions.

$$C_{eng} = \sum_{i=1}^{k-1} (\| \mathbf{m}'_i - \mathbf{m}'_{i+1} \|)^2 \tag{9}$$

Instead of evaluating the full mesh of discrete energies through steps, we simplify the computation into a single pass method:

$$C_{eng_{ij}} = C'_{eng_{i-1}} + (\| \mathbf{m}'_{ij} - \mathbf{m}'_{(i-1)j} \|)^2 \tag{10}$$

where $C'_{eng_{i-1}} = min(C_{eng_{(i-1)j}})$.

C. Iterative Plan Adjustment (IPA)

Recall that we have defined F_x as the features denoting waypoints in a more complex situation. After the *task* path is generated, we will then check if any $\mathbf{m}'$ contradicts with $\mathbf{b}_i$ in F_x. Suppose we define the contradiction evaluation as $Ct(F_x)$:

$$Ct(F_{xi}) = \begin{cases} 1 & \text{for } \mathbf{b}_i \neq \mathbf{m}' \text{ if } B_i \text{ denotes a waypoint} \\ 0 & \text{for no contradiction} \end{cases} \tag{11}$$

The algorithm should iterate the following steps until all $Ct(F_{xi}) = 0$:

1) Compute $Ct(F_{xi})$ for all i
2) For any $Ct(F_{xi}) = 1$, locate the point on the *task* path that is nearest to $\mathbf{b}_i$. Find the corresponding point $\mathbf{a}_j$ in the *template* path. Put $\mathbf{a}_j$ into set F and $\mathbf{b}_i$ into F'.
3) Re-perform the distortion mapping algorithm described above to find a new *task* path.

V. EXPERIMENTS

We implemented and validated our planning algorithm on the visual sensors of a humanoid robot, the iCub (Fig. 1a), developed by the RobotCub Consortium[1].

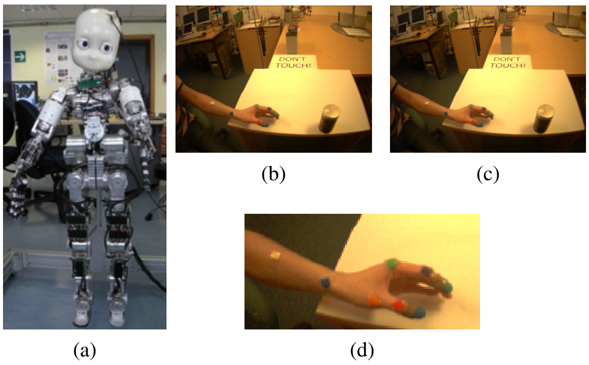


Fig. 1: **The experiment set-up for testing the path planning algorithm.** The iCub in (a) is developed by the RobotCub Consortium. It has a total of 53 Degrees of Freedom, 32 of which are distributed on the arms. (b) and (c) are an instance of a human subject with markers captured by the left and right cameras of the iCub respectively.(d) shows the locations of markers placed on the left arm of the human subjects.

A. Experimental Setup

We made use of the pair of stereo cameras on-board as the means of obtaining the demonstrated information. In the following reported experiments, the iCub captured the demonstrations at the frame rate of 20Hz and frame resolution of 320×240 pixels/camera (an example is shown in Fig. 1b & 1c).

As the means of extracting the points of interest present in a scene, in this case, the location of relevant joints of a human arm, is a computer vision problem on its own, we bypassed this issue by placing markers on the human subject (e.g. Fig. 1d) to be tracked. A human demonstrator was instructed to perform a grasp-oriented task while the iCub observed the actions. Then the iCub was given a new location of the arm and the invariant feature matrices in an attempt to produce a desired path.

B. Experiment Descriptions

A total of 5 different experiments have been conducted. Each experiment consists of 15 trials performed by different subjects. This has resulted in 75 different experiments available for cross-validation of the model with the human demonstrations.

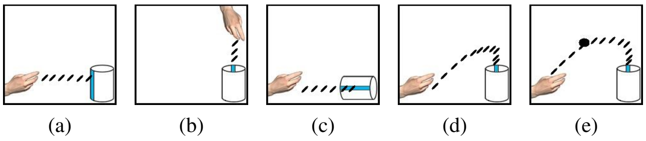


Fig. 2: **The diagramatics of the 5 conducted experiments.** All subjects have been requested to use their inferior arm (in all cases, left arms). The hand positions in the diagrams indicate the starting points of the experiments. The subjects have also been requested to approach the object with their forearms orthogonal to the blue strips indicated in the diagrams. 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.

[1] www.RobotCub.org

The 5 experiments were designed to test the robustness of the proposed model in various ways, which are described below and illustrated in Fig. 2.

1) Experiment 1 shown in Fig. 2a is the most general case of grasping, hypothesised to be most useful for mapping in complex situations.
2) Experiment 2 shown in Fig. 2b is intended to test the algorithm with only pure angular rotation of the entire scene.
3) Experiment 3 shown in Fig. 2c is designed to test the ability of generalisation of the algorithm into a 3D situation.
4) Experiment 4 shown in Fig. 2d is to test the general performance of the warping algorithm.
5) Experiment 5 shown in Fig. 2e is to test the robustness of the algorithm in a more complex situation, i.e. in this case with a waypoint.

C. Implementation of the Algorithm

As we can see from Fig. 1b & 1c, both cameras on-board the iCub have certain extent of fish-eye distortion. Thus, before processing the captured frames, we undistorted the images with a set of calibration parameters discussed in [20].

We extract the marker positions by using an efficient colour segmentation technique proposed by Bruce et al [21]. It can be inferred that there are noise and uncertainty present in the image and the extraction process. As limited by the resolution of the image, any sub-pixel information cannot be recorded. Assuming we have full accuracy in extracting the markers, the least amount of uncertainty involved is therefore 0.5 pixel. Thus, we applied Gaussian Radial Basis Function (RBF) smoothing with smoothing parameter of 0.5 pixels to the extracted path from the demonstrations.

We then set up the algorithm with intrinsic parameter $\lambda = 0$ as we believe that the RBF smoothing should have helped to remove the noise present in the experiments. Thus, any attempt to relax λ might result in the distortion of the optimal path. We also thresholded the distortional energy to accommodate the inclusion of pixel locations up to 2 pixels away from the mapped sub-pixel location. The final step is to use the intrinsic parameter of the stereo cameras to generate the 3-D action path.

D. Performance Evaluation

Although one can use visual inspection to spot paths that match the input, however, to evaluate the performance of the proposed algorithm quantitatively, we introduce three performance metrics as measures of success, namely absolute path difference, mean squared difference and correlation coefficient of paths.

1) Absolute Path Difference (APD): Physically, APD measures the difference between the two paths in comparison in terms of distance.

$$APD = \left| \sum_{i=1}^{N-1} \| \mathbf{m}'_{i+1} - \mathbf{m}'_i \| - \sum_{i=1}^{N-1} \| \mathbf{m}_{i+1} - \mathbf{m}_i \| \right| \tag{12}$$

2) Mean Squared Difference (MSD): In statistics, this quantity is commonly known as the Mean Squared Error. We make use of MSD to estimate the variance between the calculated path and the demonstrated path to guage how closely the algorithm is to human path planning.

$$MSD = \frac{1}{N} \sum_{i=1}^{N} \| \mathbf{m}'_i - \mathbf{m}_i \| \tag{13}$$

3) 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.

$$R^2 = \frac{\sum_{i=1}^{N} (\mathbf{m}_i - \bar{\mathbf{m}}) \cdot (\mathbf{m}'_i - \bar{\mathbf{m}}')}{\sqrt{(\sum_{i=1}^{N} (\mathbf{m}_i - \bar{\mathbf{m}})^2)(\sum_{i=1}^{N} (\mathbf{m}'_i - \bar{\mathbf{m}}')^2)}} \tag{14}$$

As we can see from (12)-(14), all the performance metrics require the input vectors to be of the same lengths. However, in actual practice, we cannot ensure all demonstrations to be completed at the same duration. Thus, we employed the Cubic Spline Interpolation method to lengthen the path with fewer waypoints to match that of the longer one.

VI. RESULTS AND DISCUSSIONS

Based on the 75 experiments we gathered from different demonstrators, we performed cross-validation of the experimental trials, i.e. for each demonstrated path,we iterated the algorithm through all the 75 possible scenarios, including the input scenario itself, to generate a 75×75 matrix of performance indicators.

Firstly, we evaluated the performance of the algorithm on the ability to preserve the original path. This is indicated by the 75 self-mapping cases. The performance metrics are tabulated in TABLE I. All 5 experiments have a confidence indicator of greater than 99% that the mapped paths preserve the input ones with very low APD and MSD. The small discrepancies shown in TABLE I are likely due to the smoothing and rounding of the paths during computation.

To understand how well the paths mapped from trials of a particular experiment can be generalised into other experiment scenarios, we grouped the performance indicators according to input/output experiments and take the mean of these indicators. The results are shown in TABLEs II, III and IV.

By examining the R^2 shown in TABLE IV, it can be easily spotted that the proposed algorithm is capable of generating paths that are close enough to what human demonstrated with 88% of the cases having R^2 of greater than 0.7. From TABLEs II and III, we can see that mapping a less complex path, e.g. straight line case or cases without forced waypoints, to any given scenario results in closer performance to that of human. Such observation is somewhat expected due to the lack of extra invariant information in the complex case. It can also be inferred that mapping of the demonstrated path to experiments with similar scenario, such as Experiment 4

TABLE I: Performances of the algorithm for all self-mapping cases.

Indicators	Exp 1	Expt 2	Expt 3	Expt 4	Expt 5
APD	2.59	2.53	3.46	4.64	6.79
MSD	24.6	10.7	41.2	38.6	27.0
R^2	0.996	0.999	0.991	0.991	0.994

TABLE II: The averaged Absolute Path Difference for mapping from one experiment to another. Columns indicate input while rows indicate output.

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Exp 1	4.0	3.4	36.9	188.9	249.4
Exp 2	4.8	3.2	88.2	111.4	179.6
Exp 3	13.5	75.7	11.0	19.2	106.8
Exp 4	46.2	21.6	43.6	9.8	18.0
Exp 5	36.3	37.3	55.8	22.5	9.5

TABLE III: The averaged Mean Squared Difference for mapping from one experiment to another. Columns indicate input while rows indicate output.

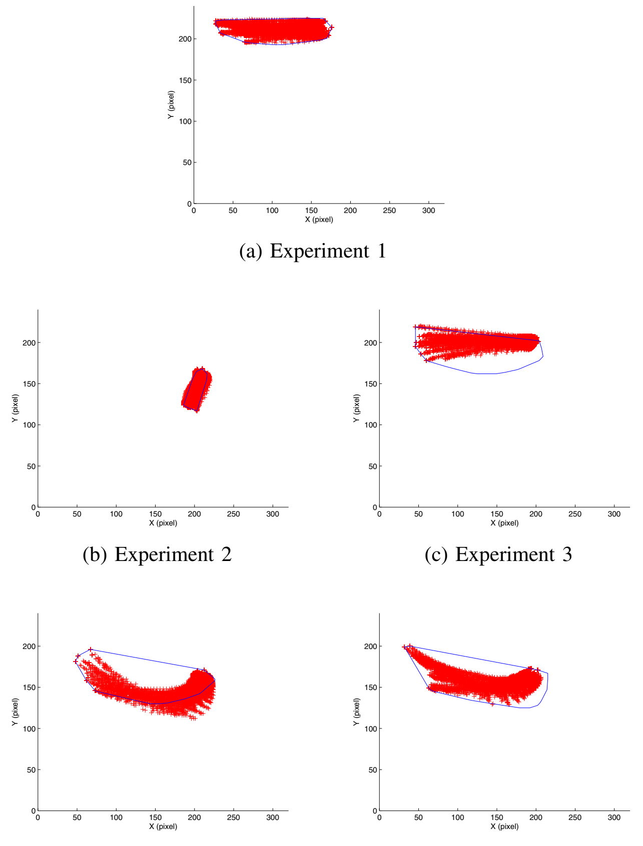
	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Exp 1	176	298	1222	7388	8390
Exp 2	117	57	1899	2809	5095
Exp 3	799	2832	399	962	2869
Exp 4	1548	667	1164	203	343
Exp 5	1139	834	1286	459	280

TABLE IV: The averaged Correlation Coefficients for mapping from one experiment to another. Columns indicate input while rows indicate output.

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Exp 1	0.961	0.953	0.817	0.414	0.380
Exp 2	0.993	0.995	0.941	0.740	0.711
Exp 3	0.861	0.444	0.891	0.774	0.747
Exp 4	0.757	0.817	0.871	0.957	0.938
Exp 5	0.835	0.816	0.872	0.885	0.962

and Experiment 5, can yield similar results although some defining conditions are different. From these results, we can believe that utilisation of this One-Shot Learning algorithm can reduce the cost and burden of repeated demonstrations, while maintaining a high level of mapping accuracy.

Fig. 3 shows examples for each mapped experiment. We can see that in most cases, the generated waypoints are bounded by the convex hull around the human demonstrated paths. In Fig. 3e, the paths generated by the IPA method are also well bounded by its demonstrated convex hull. This implies that the IPA algorithm works in dealing with scenes with extra information. Thus, we believe that with similar strategy, obstacle avoidance can also be carried out with this algorithm by carefully moving the path point to the neighbour of the obstacle with lowest bending energy. As such, this algorithm can also be treated as an incremental learning paradigm in path planning to accommodate additional constraints. When extra information, such as addtional waypoints and obstacles, appears in the planned route during action, the algorithm can make use of the IPA to generate an adapted path from the current location. Potentially, as this algorithm treats the movement of each point-of-interest independently, agent-based approach can be applied to more complex tasks, such as coordinated bi-manual operations by specifying the constraints between agents.



(a) Experiment 1

(b) Experiment 2

(c) Experiment 3

(d) Experiment 4

(e) Experiment 5

Fig. 3: **Examples of the resulted paths for the 5 conducted experiments.** The blue polygons denote the bounding area of all the paths that users have demonstrated for an experiment. The red scatter plots denote the paths generated for an experiment from the cross-validation of all experimental trials.

Inevitably, we can see that there are cases that the algorithm does not reproduce a good path most likely due to the quality of the demonstrated path. However, we can tackle such problems by adding a secondary layer to this algorithm to classify and generalise from those templates that are repetitions of each other to generate a more general template. On the other hand, as the templates are stored as a set of path waypoints and invariant features, users will have the flexibility to understand the underlying contexts and make necessary adjustments such as removing underperformed templates such as those that have low R^2. This is particularly useful as it does not require retraining of the model.

VII. CONCLUSIONS

In this paper, a One-Shot Learning robot path imitation algorithm has been presented. This algorithm has been implemented and cross-validated using the 75 sets of experimental data conducted on human subjects, captured by the stereo cameras of the iCub humanoid robot. The experimental results show that this generic algorithm is capable of reproducing satisfactory path by imitating simple tasks. However, the experiments have been conducted with assumptions. For example, it has been assumed that sufficient invariant feature points have been given for mapping and the destination features are static. Computationally, this algorithm with the implementation of the approximation method is an O(n^3) problem. As compared to an imitation algorithm that focuses solely on accuracy by using fluid dynamics principles[22], this algorithm preserve a good level of accuracy but is much computationally inexpensive.

We plan to extend a weighted approach towards template selection of this algorithm with self-experimentation of the robot. This is to reduce the need of human intervention in removing undesirable templates by introduction of weights to the templates, once more than one template of the same task is observed.

VIII. ACKNOWLEDGMENTS

The authors gratefully ackowledge the invaluable comments and continued support from the members of the BioART Lab - Simon Butler, Tom Carlson, Murilo Fernandes Martins, Balint Takacs and Paschalis Veskos, especially to Bálint Takács for many extended and insightful discussions. The authors would also like to express their gratitude towards the participants in the experiments.

REFERENCES

[1] C. Nehaniv and K. Dautenhahn, “Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications,” *World Scientific Series in Robotics and Intelligent Systems*, vol. 24, pp. 136–161, 2000.

[2] 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.

[3] S. Muench, J. Kreuziger, M. Kaiser, and R. Dillman, “Robot programming by demonstration (RPD)-Using machine learning and user interaction methods for the development of easy and comfortable robot programming systems,” in *Proceedings of the International Symposium on Industrial Robots*, vol. 25, pp. 685–685, 1994.

[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*, pp. 1–20, 2003.

[5] Y. Yoshikawa, K. Shinozawa, H. Ishiguro, N. Hagita, and T. Miyamoto, “Responsive robot gaze to interaction partner,” in *Proceedings of robotics: Science and systems*, 2006.

[6] 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.

[7] A. Ude, C. Atkeson, and M. Riley, “Programming full-body movements for humanoid robots by observation,” *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 93–108, 2004.

[8] 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.

[9] M. Nicolescu and M. Mataric, “Natural methods for robot task learning: Instructive demonstrations, generalization and practice,” in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 241–248, ACM, NY, USA, 2003.

[10] T. Mitchell, R. Keller, and S. Kedar-Cabelli, “Explanation-based generalization: A unifying view,” *Machine learning*, pp. 47–80, 1986.

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

[12] H. Friedrich, R. Dillmann, and O. Rogalla, “Interactive robot programming based on human demonstration and advice,” *Lecture notes in computer science*, pp. 96–119, 1999.

[13] 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.

[14] B. Takács and Y. Demiris, “Multi-robot plan adaptation by constrained minimal distortion feature mapping,” in *IEEE International Conference on Robotics and Automation, ICRA ’09*, May 2009.

[15] D. Lowe, “Object recognition from local scale-invariant features,” in *Int’l Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999.

[16] F. Bookstein, “Principal warps: Thin-plate splines and the decomposition of deformations,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 6, pp. 567–585, 1989.

[17] M. Powell, “A thin plate spline method for mapping curves into curves in two dimensions,” *Computational Techniques and Applications (CTAC ’95)*, 1995.

[18] J. Zhu and M. Lyu, “Progressive finite newton approach to real-time nonrigid surface detection,” in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR’07*, pp. 1–8, 2007.

[19] J. Pilet, V. Lepetit, and P. Fua, “Real-time nonrigid surface detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 1, 2005.

[20] 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*, vol. 0, p. 1106, 1997.

[21] J. Bruce, T. Balch, and M. Veloso, “Fast and inexpensive color image segmentation for interactiverobots,” in *2000 IEEE/RSJ Int’l Conference on Intelligent Robots and Systems(IROS 2000). Proceedings*, vol. 3, 2000.

[22] H. Mayer, I. Nagy, A. Knoll, E. Braun, R. Lange, and R. Bauernschmitt, “Adaptive Control for Human-Robot Skilltransfer: Trajectory Planning Based on Fluid Dynamics,” in *2007 IEEE International Conference on Robotics and Automation*, pp. 1800–1807, 2007.