

An Intuitive Robot Learning from Human Demonstration

Uchenna Ogenyi, Zhaojie Ju, and Honghai Liu

School of Computing,
University of Portsmouth,
Portsmouth, United Kingdom
uchenna.ogenyi@port.ac.uk
zhaojie.ju@port.ac.uk
honghai.liu@port.ac.uk

Abstract. This paper presents a new way to teach a robot certain motions remotely from human demonstrator. The human and robot interface is built using a Kinect sensor which is connected directly to a remote computer that runs on processing software. The Cartesian coordinates is extracted, converted into joint angles and sent to the workstation for the control of the Sawyer robot. Kineshetic teaching was used to correct the reproduced demonstrations while only valid resolved joint angles are recorded to ensure consistence in the sent data. The recorded dataset is encoded using GMM while GMR was employed to extract and reproduce generalised trajectory with respect to the associated time-step. To evaluate the proposed approach, an experiment for a robot to follow a human arm motion was performed. This proposed approach could help non-expert users to teach a robot how to perform assembling task in more human like ways.

Keywords: Programming by Demonstration, Learning from Demonstration, Learning by Imitation.

1 Introduction

Since the introduction of collaborative robots, humans can currently work in the same workspace with robots and the application of robot in human endeavour has continued to increase. However, the applicability of robot will be much diversified if it is much easier to teach a robot how to perform a task. The most viable technique that is currently used to teach a robot how to perform a task is Programming by Demonstration (PbD) which is also known as Learning from Demonstration (LfD) or Learning by Imitation. This technique enables a robot to learn from human teacher or other robots. The goal is to let users program a robot by simply giving it instructions on what to do in the form of demonstrations or by using other interactive means [9], [4]. This method, of course reduces the cost of employing professionals to program the robot as it consequently allows the robot to learn from the users regardless of the users knowledge of robotics or robot programming.

In robotics, many researchers have employed various PbD techniques in order to transfer human skills to a robot. This includes the use of optical marker to track the

motion of the human demonstrator [3]. This process involves fixing a tracking sensor on the human body to measure the skeletal movement of the part of the body of interest. However, the equipment causes inconvenience to the human user and sometimes results to data drift. Similar but more convenient technology was employed in [11] to control a robot via teleoperation. In the paper, the authors employed the use of a vision sensor (Kinect) to capture human joint information and then directly mapped it to the robot manipulator in order to control the robot in real time. By using this approach, human is free from the inconveniences that arise in the former. Another interesting approach is kinesthetic teaching which involves guiding the robot by the operator by moving the robot arm through the task to be implemented [5]. Kinesthetic teaching holds some advantages over other aforementioned approaches. Firstly, Kinesthetic approach does not resolve to singularity and external hardware devices that may cause inconveniences to human demonstrator are not required. Secondly, data recording more complete and accurate as a human physically guides the robot through the desired path or task. Some other approaches have been used and some have involved combining two approaches in other to utilize the advantages that both could offer. In [6], both kinesthetic teaching and the use of motion sensor were employed in the proposed human to robot gesture skill transfer.

In this paper we proposed a method that employed similar approach with [6] but with a different strategy. Unlike in [6] were both Kinesthetic teaching and motion sensor that could cause data drift and inconvenience to users were employed, in this paper, both kinesthetic teaching and vision based sensor which does not require users to wear any additional interface and permits for better orientation of the human arm were employed. During the demonstration of task, the Kinect sensor allows the user to record the human arm movement which is mapped directly to the robot controller to allow the robot reproduce the demonstrated gesture. On the other hand, Kinesthetic teaching is then applied to help the robot arm move correctly to the desired posture. During the process, valid trajectories from the demonstrations were extracted and thereafter Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR) were employed to encode and retrieve the generalized version of the trajectories respectively.

2 System Description

The framework comprises of two computer systems where one (remote) is connected directly to a Kinect sensor and the second (workstation) is connected to the Sawyer robot. Teaching the robot starts when the demonstrator stands in front of the Kinect sensor and maintains a certain distance and line of sight with the Kinect sensor. The human demonstrator moves his right arm to demonstrate sequences of steps to follow for the workings of the task to be taught. Then the Kinect sensor tracks the motion of the human arm and allows for human joint positions to be extracted. Employing vector approach [12], joint angles of the robot arm was calculated and passed to the workstation connected to Sawyer robot via User Datagram Protocol (UDP). By this means, human joint position is mapped to the robot arm and the robot arm is made to move by mimicking the human arm.

2.1 Hardware Setup

The platform for this research is centered on Sawyer robot. Sawyer is a single-arm robot with a total of eight degrees of freedom (8DOF) consisting of joint0 to joint6 (j0-j6) on the arm and one more joint on the head (see Fig. 1). Apart from j6 which has 540-degree of rotation that enables smooth positioning of the end-effector in different angles and also increase smooth maneuvering around obstacles; the rest of the joints have 350-degree of rotation. Sawyer is embedded with several sensors and actuators that are essential for research. Sawyer is provided with a motor encoder for every DOF, except for the head actuator. This makes it possible for joint angles to be extracted from each joint. Also embedded are force/torque sensor on each of its joints which accounts for collision detection and a Cognex camera attached to the wrist close to the end-effector which could be used to ensure a better view of the object it interacts with. The Robotiq [1], recorded that the high precision of Sawyer is because the harmonic drive motors comes with zero backlash gear boxes.

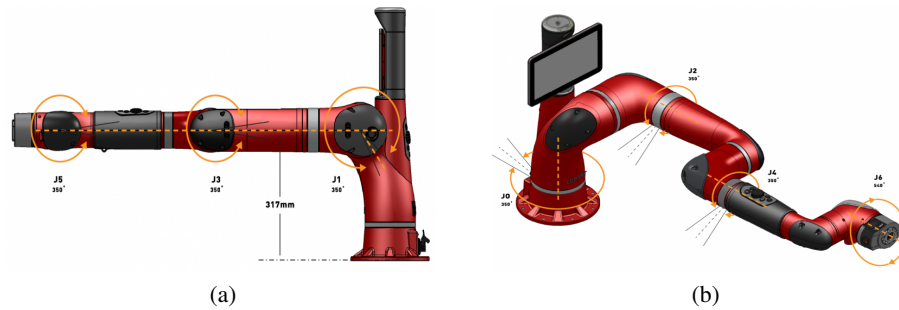


Fig. 1: The figures shows Sawyer robot joints and link lengths: (a) Show the Pitch joints while (b) Shows all the joints and their corresponding link lengths

The Kinect sensor can provide both RGB and depth information of a human teacher. The teacher performs demonstrations which is teleoperated to the robot via a Kinect sensor. During this process, human arm joint positions are mapped to the robot arm and the joint angle trajectories are recorded while robot arm is being moved by mimicking human motion.

2.2 Software Components

Processing software :Processing software is an open source software that was built for teaching fundamentals of programming language. Processing comes with very simple syntax and a lot of graphical visual feedback to make programming easier for non-programmers. Processing is compatible with Windows and Linux and comes with a lot of libraries and functions that enables integration with various devices such as Kinect sensor.

Robot Operating System and ROSPY : Sawyer robot runs on Robot Operating System (ROS) platform. ROS has a lot of software libraries and tools that allows for development of robotic projects. Various research-oriented robots including Sawyer use

this platform for the development of various kinds of robotic projects. With rospy, programmer can rapidly interface python with the ROS tools and parameters. Rospy pays a great interest on the developers' time over runtime which in turn creates enabling environment for speedy prototyping and testing of algorithm within ROS [2].

Transmission Protocol : The traditional Transmission Control Protocol (TCP) ensures no packet lost during data transmission but does this at the merit of slower transmission speed [10]. User Datagram Protocol (UDP) became a better candidate since fast speed of data transmission is of paramount importance in this system. In this paper, the UDP was used to transmit data from the remote computer to the workstation which is connected directly to the sawyer robot.

3 Probabilistic Model Selection

Several statistical models have been used to successfully model human motion and retrieve generalised version of the motion in the past. However, Gaussian Mixture Model (GMM)/Gaussian Mixture Regression (GMR) is less sensitive to noisy data and computationally inexpensive as it has a fast convergence rate. In this paper, the encoding, generalisation and reproduction of the trajectories will be performed using GMM/GMR.

3.1 Gaussian Mixture Model

Considering a dataset of N data points with a dimensionality of D collected when a human demonstrator performed a task. Each sensory data gathered during the demonstrations contain joint angles and corresponding time-step. Therefore, the joint angle trajectories collected are $\chi = \{\chi_j\}_{j=1}^N$. Each datapoint consist of temporal and spatial variables which are donated by t and s as in $\chi = \{\chi_t, \chi_s\}_{j=1}^N$ and has χ_t and χ_s , for the temporal and spatial vectors respectively.

The dataset was used to encode the GMM of K components as defined by the probability density function:

$$p(\chi_j) = \sum_{k=1}^K p(k)p(\chi_j|k) \quad (1)$$

where $p(k)$ is the prior and $p(\chi_j|k)$ is a functional probability density function. For a mixture of K Gaussian distributions of dimensionality D , the parameters in equation (1) are defined as:

$$\begin{aligned} p(k) &= \pi_k \\ p(\chi_j|k) &= N(\chi_j; \mu_k, \Sigma_k) = \\ &= \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \ell^{\frac{((\chi_j - \mu_k)^T \Sigma_k^{-1} (\chi_j - \mu_k))}{2}} \end{aligned} \quad (2)$$

where $N(\chi_j; \mu_k, \Sigma_k)$ represents the probability of a datapoint χ with respect to the normal distribution $N(\mu, \Sigma)$ and the Gaussian distributions defined by the prior probability π_k , mean vectors μ_k and covariance matrices Σ_k .

3.2 Gaussian Mixture Regression

After encoding the trajectories, GMR was employed to retrieve a smooth generalized version of the trajectories. Using GMR retrieval process is more advantageous over other stochastic methods as it can provide a more reliable way of reconstructing the Gaussian model [7]. The observed data $\chi = \{\chi_t, \chi_s\}$ is first modelled by the joint probability distribution $P\{\chi_t, \chi_s\}$. A generalized trajectory is then computed by estimating $E[p(\chi_s|\chi_t)]$ and $cov[p(\chi_s|\chi_t)]$ which is used to extract the constraints of the performed task. Just like in a regression problem, where a set of input variable $X \in \mathbb{R}^p$ and response variable $Y \in \mathbb{R}^q$ are given, where p and q are the dimensionality of the model input and output respectively. The aim of the regression, is to estimate the conditional expectation of Y given X from a set of observations $\{X, Y\}$. In the case of this paper, the regression aims at estimating the conditional probability of χ_s given χ_t where χ_s is a vector of positions at a given time χ_t . Therefore, by computing the conditional expectation of χ_s at each time step, the generalized trajectories are obtained.

$$\mu_k = \{\mu_{t,k}, \mu_{s,k}\}, \Sigma_k = \begin{pmatrix} \Sigma_{tt,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{ss,k} \end{pmatrix}$$

For each component of k , the expected distribution of $\chi_{s,k}$ given temporal value χ_t is

$$\begin{aligned} p(\chi_{s,k}|\chi_{t,k}) &= N(\chi_{s,k}; \hat{\chi}_{s,k}, \hat{\Sigma}_{ss,k}) \\ \hat{\chi}_{s,k} &= \mu_{s,k} - \Sigma_{st,k}(\Sigma_{tt,k})^{-1}(\chi_t - \mu_{t,k}) \\ \hat{\Sigma}_{ss,k} &= \Sigma_{ss,k} - \Sigma_{st,k}(\Sigma_{tt,k})^{-1}\Sigma_{ts,k}, \end{aligned}$$

The mixture of K component distributions $N(\hat{\chi}_{s,k}, \hat{\Sigma}_{ss,k})$ is done according to the prior β_k , where $\beta_k = p(k|\chi_t)$ is determined by the probability of the component k .

$$p(\chi_s|\chi_t) = \sum_{k=1}^K \beta_k N(\chi_s; \hat{\chi}_{s,k}, \hat{\Sigma}_{ss,k}) \quad (3)$$

$$\beta_k = \frac{p(k)p(\chi_t|k)}{\sum_{i=1}^K p(i)p(\chi_t|i)} = \frac{\pi_k N(\chi_t; \chi_{t,k}, \Sigma_{tt,k})}{\sum_{i=1}^K \pi_k N(\chi_t; \chi_{t,i}, \Sigma_{tt,i})}$$

Using equation (3) an estimation of the conditional expectation χ_s given χ_t is computed for a mixture of K components as show in equation (4). A generalized form of the trajectory and associated covariance matrix as used to reproduce the movement by evaluating the $(\hat{\chi}_s, \hat{\Sigma}_{ss})$ at different time steps χ_t .

$$\hat{\chi}_s = \sum_{k=1}^K \beta_k \chi_{s,k}, \quad \hat{\Sigma}_{ss} = \sum_{k=1}^K \beta_k^2 \hat{\Sigma}_{ss,k}. \quad (4)$$

3.3 Learning Parameter

Expectation Maximization algorithm as proposed in [8] was adopted and used to compute the parameter of the Gaussian mixture model. Given that $p_{k,j}$ is defined as the posterior probability $p(k|\chi_j)$ computed using Bayes theorem $p(k|\chi_j) = \frac{p^{(k)}p(\chi_j)}{\sum_{i=1}^K p^{(i)}p(\chi_j|i)}$. Using a rough estimation by k-means segmentation, the parameter $\theta = \pi_k, \mu_k, \Sigma_k, E_k$ are iteratively computed until convergence.

E – step

$$p_{k,j}^{(r+1)} = \frac{\pi_k^{(r)} N(\chi_j; \mu_k^{(r)}, \Sigma_k^{(r)})}{\sum_{i=1}^K \pi_i^{(r)} N(\chi_j; \mu_i^{(r)}, \Sigma_i^{(r)})}$$

$$E_k^{(r+1)} = \sum_{j=1}^N p_{k,j}^{(r+1)}$$

M – step

$$\begin{aligned} \pi_k^{(r+1)} &= \frac{E_k^{(r+1)}}{N} \\ \mu_k^{(r+1)} &= \frac{\sum_{j=1}^N p_{k,j}^{(r+1)} \chi_j}{E_k^{(r+1)}} \\ \Sigma_k^{(r+1)} &= \frac{\sum_{j=1}^N p_{k,j}^{(r+1)} (\chi_j - \mu_k^{(r+1)}) (\chi_j - \mu_k^{(r+1)})^T}{E_k^{(r+1)}} \end{aligned}$$

4 Experimental Results

For each given gesture, 3 demonstrations involving the use of the vision sensor and 3 demonstrations using Kinesthetic teaching are provided. Thereafter, the original dataset of 7DOFs was reduced to 2-dimension but the important variance is retained. With this, the GMM parameter is estimated using the EM algorithm. From Fig.2b, it could be observed that 3-5 GMM components could fit the encoded gestures. Using GMR a generalized trajectory among the demonstrated trajectories could be retrieved as presented in Fig.2c.

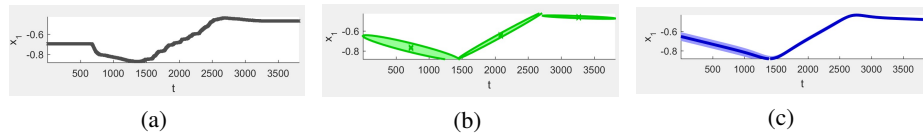


Fig. 2: (a) A set of sample demonstrations performed by a human. (b) The computed GMM for the demonstrated task (c) The GMR reproduced trajectory of the tasked demonstrated.

Reproduction of the 3 demonstrated gestures by the Sawyer robot was evaluated as shown in Fig.3. The resulting trajectories of the robot correspond to the motion performed by the demonstrator. The trajectories of the robot represent the joint angles generated during task reproduction. Due to technical limitations, the reproduced demonstration by the robot is not exactly the posture of the demonstrator (see Fig.3).

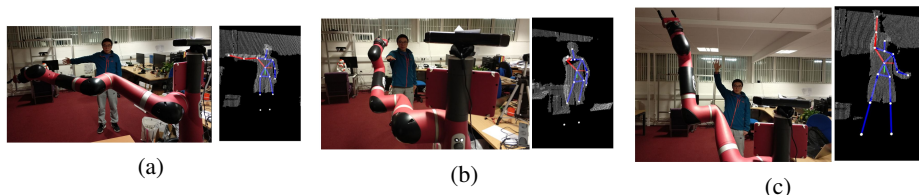


Fig. 3: The robot mimics the resulting trajectory of the human arm motion: (a) A gesture towards the right (b) A gesture towards the robot (c) A gesture pointing upward

Another proof of consent conducted was asking some group of people to collaborate with the robot in a box stacking task. The experiment involved 10 persons with each allowed to stack 4 blocks together in not more than five trials. Five of them practiced how to perform the block stacking task while the other half were only instructed on how to perform the task and were never given a change to practice it before performing the task. It is expected that those who practiced before performing the task should finish much earlier than those who did not, but it turns out that the time difference is not as much as anticipated. The outcome of the experiment shows that the average completion time for trained persons is 13.86 min while that for untrained persons is 26.3 min as presented in Table 1. This result is an indication that with little practice, users can perfect on the use of the proposed system. Figure 4 shows the snapshot of the stacking task performed by one of the users.

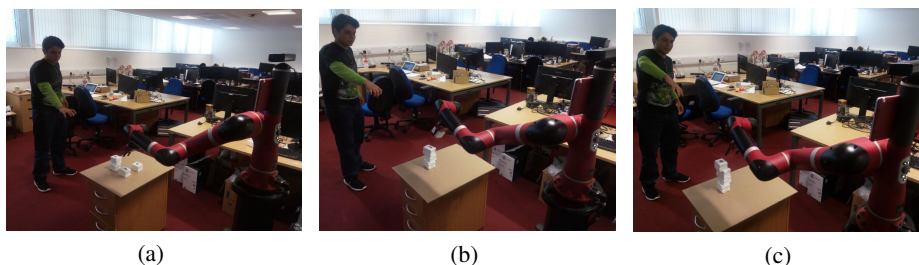


Fig. 4: The task was to stack the boxes in the order of 1-4. Figures (a), (b) and (c) are snapshots of a user performing the stacking task.

Table 1: Block stacking result

Test Category	Trained Person	Untrained Person
Stack completion time	13.86 ±2.729	26.3 ±3.976

5 Conclusion

This paper shows a new way to teach a robot how to perform a task via Programming by Demonstration. The approach proposed how to teach a robot to follow human arm motion based on GMM/GMR. The process commenced by acquiring the robot joint angles when it mimics human demonstrated gesture captured via a Kinect sensor. The acquired dataset is complemented using kinesthetic teaching aimed to correctly move the robot arm towards achieving a more accurate demonstration. Encoding of the recorded valid joint angles was performed using GMM while a smooth generalised trajectory of the demonstrated motion was achieved using GMR. The experiment shows that this dual means of data acquisition could be employed in training a robot via PbD.

References

1. COLLABORATIVE ROBOT EBOOK and COMPARATIVE CHART : 19 collaborative robots compared and analysed in this ebook. <http://blog.robotiq.com/collaborative-robot-ebook>, accessed: 2016-12-31
2. Package Summary, rospy. <http://wiki.ros.org/rospy>, accessed: 2017-03-23
3. Bestick, A.M., Burden, S.A., Willits, G., Naikal, N., Sastry, S.S., Bajcsy, R.: Personalized kinematics for human-robot collaborative manipulation. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 1037–1044. IEEE (2015)
4. Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot programming by demonstration. In: Springer handbook of robotics, pp. 1371–1394. Springer (2008)
5. Calinon, S., Billard, A.: Active teaching in robot programming by demonstration. In: Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on. pp. 702–707. IEEE (2007)
6. Calinon, S., Billard, A.: Incremental learning of gestures by imitation in a humanoid robot. In: Proceedings of the ACM/IEEE international conference on Human-robot interaction. pp. 255–262. ACM (2007)
7. Calinon, S., Guenter, F., Billard, A.: On learning, representing, and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37(2), 286–298 (2007)
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological) pp. 1–38 (1977)
9. Ge, J.G.: Programming by demonstration by optical tracking system for dual arm robot. In: Robotics (ISR), 2013 44th International Symposium on. pp. 1–7. IEEE (2013)
10. Masirap, M., Amaran, M.H., Yussoff, Y.M., Ab Rahman, R., Hashim, H.: Evaluation of reliable udp-based transport protocols for internet of things (iot). In: Computer Applications & Industrial Electronics (ISCAIE), 2016 IEEE Symposium on. pp. 200–205. IEEE (2016)
11. Reddivari, H., Yang, C., Ju, Z., Liang, P., Li, Z., Xu, B.: Teleoperation control of baxter robot using body motion tracking. In: Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on. pp. 1–6. IEEE (2014)
12. Shiffman, D.: A Roadmap for U.S. RoboticsA vector is a collection of values that describe relative position in space. <https://processing.org/tutorials/pvector/>, accessed: 2018-03-20