DOI 10.31256/UKRAS19.25

Visual Features as Frames of Reference in Task-Parametrised Learning from Demonstration

Shirine El Zaatari, Weidong Li

Abstract— **Task-parametrised learning from demonstration (TP-LfD) is suitable for programming collaborative robots (cobots) for collaborative industrial tasks, since the algorithm is able to encode complex mappings between observed states to the cobot's actions. TP-LfD relies heavily on perception, since detected objects and people serve as task parameters. This is a challenge since 1) industrial objects are difficult to detect due to their irregular shapes and sizes and 2) using marker stickers for detection is not desirable in manufacturing scenarios. Moreover, another challenge of using TP-LfD is that although it is an intuitive programming method, it is difficult for operators to initialise it due to their lack of underlying theoretical knowledge as opposed to the researchers that previously tested the algorithm. We aim to address these two challenges simultaneously by building an automatic task parametriser in which reinforcement learning is used to assign task parameters from a set of randomly detected visual features. In this paper, we introduce our solution and the progress done so far.**

I. Introduction

Industrial parties are growing increasingly interested in implementing human-robot collaboration (HRC) on their shop floors. Collaborative robots (cobots) bring many benefits to manufacturing including mass customisation and improved operator working conditions. For these benefits to be attained, the cobot must have a high level of intelligence and flexibility since it will be working alongside a human. It must act according to real-time operator and object states while respecting task requirements. The cobot should ideally also be easily programmable by a non-expert operator to ensure quick deployment and adjustability. One promising algorithm is learning from demonstration (LfD), in which the operator can intuitively teach the cobot an industrial task by recording a few demonstrations of it being done. Different variants of learning from demonstration are able to capture different levels of mapping complexities between states and actions. Task-parametrised learning from demonstration (TP-LfD) is thought to be able to capture the widest range of task instances, making it a generic algorithm for learning the widest range of collaborative industrial tasks [1].

However, although many algorithms exist that enable humans to intuitively teach cobots complex tasks, these algorithms are not yet popular in the industry. There are two main reasons behind this: 1) Some of these algorithms are not as intuitive to use by operators as researchers think they are. For example, in LfD, researchers record demonstrations while understanding the underlying theory behind the algorithm. These demonstrations are more likely to yield the desirable results as opposed to ones recorded by operators. Moreover, assuming perception is going to be achieved using sticker markers, the operator must decide where to paste the markers as to avoid occlusion, respect the shape and function of the work piece, avoid redundancy and not miss key objects, etc. This is often a challenge and could yield to errors if not done well. 2) These algorithms heavily rely on cobots' perception abilities, since the cobot must localise and detect relevant objects and people in a stochastic environment. Deep learning solutions aren't useful to identify industrial objects due to the lack of large training data on industrial parts. Pasting stickers is undesirable on manufacturing products and can be impossible due to part size and shape. Therefore, in this paper, we propose a solution to address the two main problems above.

In this paper, we discuss the methodology, while highlighting the contribution of this project. Moreover, the project progress is described and the future steps are outlined.

II. Methodology

The steps of our algorithm are outlined in Figure 1. Demonstrations are done by kinesthetic teaching and recorded by logging cobot joint data (angles and torques) and an RGB recording of the scene. The RGB images are used to detect and localise objects, in order to provide input to the TP-LfD algorithm. RGB images are inputted in a perception algorithm that extracts prominent visual features, further described in Section II-B. Since a large number of visual features will be detected, they are inputted into a reinforcement learning (RL) algorithm, briefly described in Section II-C, that filters and eliminates those with a high chance of irrelevance or redundancy. This RL policy updates according to a cost function calculated based on the performance of the TP-LfD. The TP-LfD outputs a Gaussian Mixture Model (GMM), as a mapping between the task parameters, which are the visual features, and the cobot joint data. The cobot will be able to reproduce the tasks recorded after performing Gaussian Mixture Regression (GMR) on the trained model.

A. *TP-LfD for Industrial Tasks*

Human-robot collaboration is identified as instances when the human and the cobot are working in close proximity without a barrier. A list of industrial human-robot collaboration tasks was identified and categorised (Figure 2) as follows:

- Independent: industrial applications done by the cobot after receiving instruction from the human, e.g.

Shirine El Zaatari is with the Institute of Advanced Manufacturing and Engineering at Coventry University, Coventry, UK (e-mail: elzaatas@uni.coventry.ac.uk).

Professor Weidong Li is with Engineering, Environment and Computing Faculty at Coventry University, Coventry, UK(e-mail: aa3719@coventry.ac.uk).

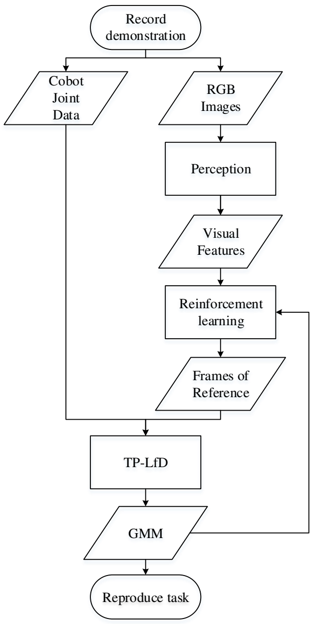

Figure 1 Flowchart showing our algorithm.

drilling, pug-in-hole, screwing, tightening bolts, surface finish, etc… The human and the cobot work in close proximity.

- Simultaneous: the cobot and the human perform similar actions towards the same goal, e.g. pick-and-place, assembly, etc... In this situation, both the cobot and the human can perform the set of actions, but distribute the actions in an optimised way according to time and space constraints.
- Sequential: the cobot performs sequential actions with the human, towards the same task goal, e.g. fetch, handover, pick-and-place, assembly, etc… The cobot should ideally understand and cater for the task needs as well as human action preference.
- Supportive: industrial applications in which the cobot is aiding the human during the task, e.g. co-manipulation, co-lifting, fixture, soldering, illumination, etc… Here, the cobot can be compliant, allowing the human to move it around. The cobot adjusts its actions according to the pace and position of the human, as well as according to task progress

In the four task categories, we identify 3 different motion types:

- Safe motion: the start and end location are known and the path is optimised/planned for collision avoidance and energy minimisation, etc.
- Compliant motion: the human is able to move the cobot manually.
- Constraint motion: the cobot follows a predefined path. This also included paths of zero length, i.e. when the cobot has to be rigid in a fixed position.

Ideally, the learning from demonstration algorithm should be able to learn the different motion types. Rozo et al. showed that TP-LfD is capable of learning the difference between compliant motion and constraint or safe motion [2]. Moreover, TP-LfD encodes variance and hence, it can differentiate between safe and constraint motion such as in [3].

TP-LfD can also capture locational and temporal constraints (motion-level actions), such as the cobot accommodating the position and the pace of the human during a supportive task [4]. This makes it suitable for supporting most of the HRC tasks in the four categories mentioned above.

Regarding task-level action decisions, e.g. subtask sequencing and scheduling, this can be modelled using Hidden Markov Model's such as in [5]. This would not interfere with the TP-LfD learnt model but rather would overlay it.

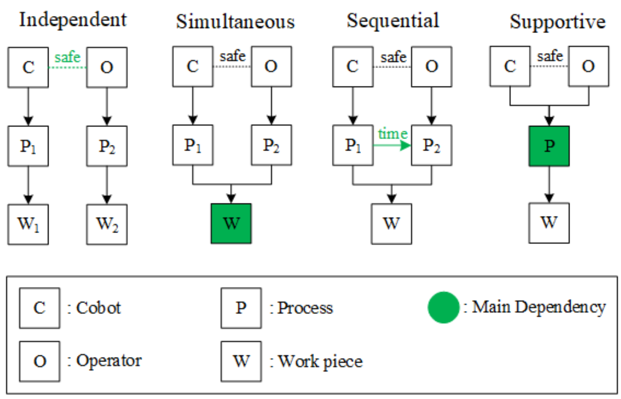

Figure 2 Categories of human-robot collaboration in industrial scenarios.

B. Reinforcement Learning on Frames of Reference

Although TP-LfD automatically calculates the relevancy of a task parameter, it will perform better if given a few task parameters of high chance of relevancy. Huang et al. were the first to tackle the problem of optimizing task parameters (frames of reference) in TP-LfD [1]. In their work, several frames of reference are initialised. Then, a reinforcement learning algorithm is used to shift the pose of these frames such that a task-specific cost function is minimised. Moreover, Huang et al. also suggest an automatic frame selection algorithm [1]. Given a number of frames, their algorithm is able to identify which frames play the biggest role in minimising the cost function. This helps eliminate frames of reference that have low influence on learning which speeds up computation and improves performance.

There are several limitations in the work of Huang et al. [1]. First, they do not tackle the question of how to visually detect frames of reference, but rather they are specified as fixed positions with respect to the cobot's end-effector. This eliminates cases in which the frames of reference are intrinsically defined on non-static objects. Second, in automatic frame selection, different subsets of frames are iteratively combined to assess different combinations of frames. This can get very computationally expensive as the number of frames increase. Third, each frame is defined as a duple $\{\mathbf{A}_t, \mathbf{b}_t\}^{(j)}$ where $\mathbf{A}_t^{(j)}$ is the rotation matrix and $\mathbf{b}_t^{(j)}$ is the translation vector of frame j at time step t. The RL algorithm updates $\mathbf{A}_t^{(j)}$ and $\mathbf{b}_t^{(j)}$, therefore, adjusting the pose of the frames of reference. This is not suitable when the frame of reference is visually detected and dynamic rather than manually specified and static.

In this project, we aim to combine a generic frame of reference detector with an automatic frame selector/optimiser. In a generic industrial task, the cost function should guarantee path optimisation and obstacle avoidance, making the cost function in [1] suitable. The update function is adjusting a relevancy score of each frame as well as a redundancy vector. The relevancy of each frame would help eliminate frames belonging to objects and locations irrelevant to the task. The redundancy vector would identify frames that belong to the same object. That way despite their relevancy, only one is incorporated in the TP-LfD. Moreover, if one frame is occluded in some portion of the demonstration, another frame from the same object/location is identified using the redundancy vector and used instead. We define our frames as a quadruple $\{\mathbf{A}_t, \mathbf{b}_t, \mathrm{r}, \mathbf{R}\}^{(j)}$ where r is the relevancy score and R is the redundancy vector, a (J-1) vector where J is the number of frames of reference. $\mathrm{R}_i^{(j)}$ is a score indicating the probability of frame i and j belonging to the same object/location.

C. *Visual Features as Task Parameters*

In TP-LfD, task parameters are usually specified as frames of reference with respect to which the cobot's motions is encoded. Task parameters can be locations in fixed space, e.g. corner of work table, or a point on an object, e.g. plate center, or action point of a tool, e.g. knife tip. These points are often marked with stickers, which are easily localised by the camera. In this project, we are looking to use visual features instead of stickers. Visual features are things like corners and edges. For a visual feature to be suitable as a frame of reference, it should satisfy certain conditions:

- Uniqueness: The feature should be relatively unique to the object/location to be localised.
- Has 6D pose: A 6D pose should be definable for the feature or through the feature.
- Easily detected and tracked: The feature should be prominent, as to be easily detected and identified from several viewpoints.

To obtain features of such characteristics, we aim to explore two main options:

1. Extracting them from feature layers of deep learning object pose estimation networks, such as Deep-6DPose [6] which takes 2D images as input.
2. Obtaining them from interest point detection deep learning networks, such as SuperPoint [7]. Figure 3 shows an example of interest points detected in a scene, extracted from [7].

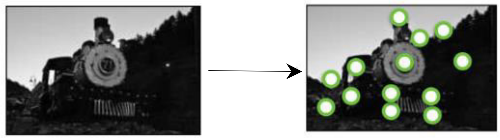

Figure 3 Example of interest point detection from SuperPoint [7].

One object can have more than one prominent feature, which if grouped together, can help solve the problem of partial occlusion.

III. IMPLEMENTATION (PROGRESS AND PLANS)

This project is divided into three main stages, reflecting the different algorithmic blocks mentioned in the methodology: 1) validating TP-LfD in industrial scenarios, 2) validating RL to automatically select frames and 3) using visual features as frames of reference.

The TP-LfD was first validated in simulation for a co-grasping task. In this task, an object was graspable from several sides. A leader agent chooses the closest side to grasp while the follower agent (the cobot), has to grasp the opposite side. This task was specifically designed to check the performance of TP-LfD on synchronised motions (the follower agent moved at a pace similar to that of the leader agent) and on conditional actions (the location of grasping of the follower agent was dependent on that of the leader agent's), which are two important attributes of HRC industrial tasks. The results, and given TP-LfD's performance on other scenarios in literature,

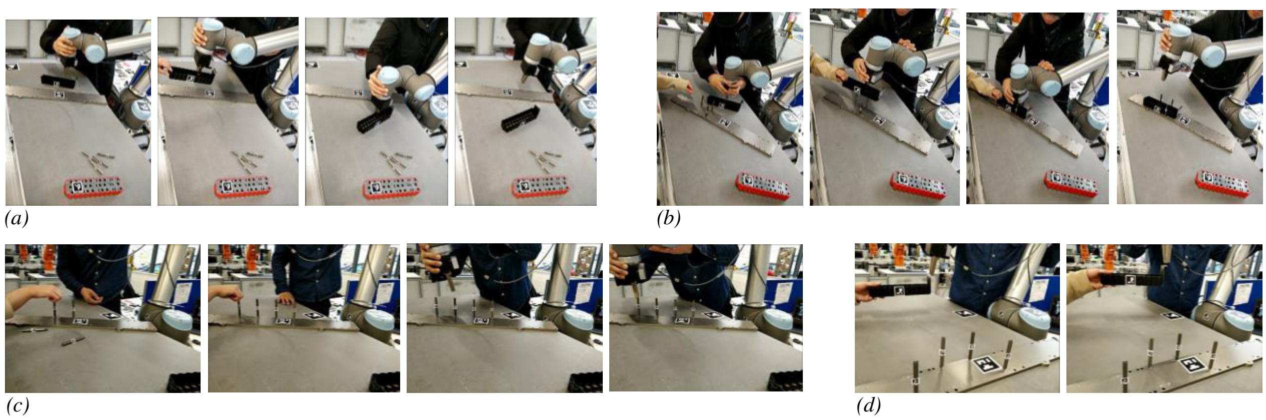

Figure 4 Examples from the recorded demonstrations of industrial sub-tasks, with battery assembly parts. (a) Pick and place from variable to variable positions. (b) Pick and place from variable to fixed positions. (c) Collaborative sequenced bolt tightening. (d) Handover task.

were satisfactory to consider TP-LfD as a promising generic programming algorithm for HRC industrial tasks.

Next, real-life training data was collected from an industrial scenario of a collaborative car battery assembly. This assembly process is usually done manually. However, it involves a few tedious tasks such as stacking battery holders, screwing bolts and carrying heavy parts. Therefore, we have chosen to explore the possibility of converting it into a collaborative task. We extracted four simple subtasks from the battery assembly process and recorded them as demonstrations:

- Pick-and-place from variable to variable positions (Figure 4(a)): In this scenario, we recorded the cobot picking up a battery box from one side of a table and placing it on the other end of the table. This is analogous to scenarios in which the cobot fetches objects for the human to work on, such a heavy battery storage boxes.
- Pick-and-place from variable to fixed positions (Figure 4(b)): In this scenario, we recorded the cobot picking up a battery box from one side of the table and sliding it into a precise location on a cooling plate. Sliding onto the plate is a constraint motion to be learnt by the TP-LfD.
- Tightening bolts (Figure 4(c)): In this scenario, the human places four bolts in random locations on a cooling plate in a random sequence. The cobot follows to touch the tip of the bolt in the same sequence. This is to learn action sequences as well as to generalise over arrayed motions.
- Handover (Figure 4(d)): In this scenario, the human changes his hand position and the cobot follows the hand. This is to teach the cobot to adjust to the human's pose and pace.

The demonstrations were recorded with sticker markers as task parameters, as an initial step in the project. In the first step of the project, we aim to learn the tasks using TP-LfD with well-placed markers as task parameters. This will give us an intuition on the performance of TP-LfD on our specific tasks. This will give further guidance on the implementation of the other project stages. The recording of the demonstration and extraction of marker poses was further proof of the challenges faced when trying to use sticker markers in an industrial scenario, namely occlusion, size limitation, and task obstruction.

In the second stage of the project, we aim to validate the RL algorithm on demonstrations with an abundance of randomly-placed markers. Third, we aim to learn from marker-less demonstrations in which visual features are the task parameters.

IV. Conclusion

Task parameterised learning from demonstration (TP-LfD) is a generic algorithm suitable for intuitively programming collaborative robots (cobots) for industrial tasks. TP-LfD requires the user to specify "frames of reference" with respect to which the cobot's motion will be encoded. These frames of reference must be relevant to the task, easy detectable and not prone to occlusion. Sticker markers have been used to specify such frames. However, using sticker markers is not desirable in industrial scenarios. Therefore, we propose the use of randomly detected visual features as frames of reference. If given a large number of frames of reference, as opposed to a select few, TP-LfD will decrease in performance. Therefore, we need to add a reinforcement learning algorithm to filter through the large set of visual features, so that only a few are passed on to the TP-LfD as frames of reference. In this paper, we present this research problem and outline our methodology, progress and future works.

Acknowledgment

Our research is supported by Coventry University, Unipart Powertrain Application ltd. and High Speed Sustainable Manufacturing Institute (HSSMI).

References

[1] Y. Huang, J. Silverio, L. Rozo and D. G. Caldwell, "Generalized Task-Parameterized Skill Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[2] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez and C. Torras, "Learning Physical Collaborative Robot Behaviors From Human Demonstrations," *IEEE Transactions on Robotics,* vol. 32, no. 3, pp. 513-527, 2016.

[3] C. Perez-D'Arpino and J. A. Shah, "C-LEARN: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[4] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics,* vol. 9, no. 1, pp. 1-29, 2016.

[5] D. Vogt, S. Stepputtis, S. Grehl, B. Jung and H. B. Amor, "A system for learning continuous human-robot interactions from human-human demonstrations," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[6] T.-T. Do, M. Cai, T. Pham and I. Reid, "Deep-6DPose: Recovering 6D Object Pose from a Single RGB Image," *arXiv preprint arXiv:1802.10367,* 2018.

[7] D. DeTone, T. Malisiewicz and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," 2017.