Physical Symbol Grounding and Instance Learning through Demonstration and Eye Tracking

Citation for published version:
Penkov, S, Bordallo, A & Ramamoorthy, S 2017, Physical Symbol Grounding and Instance Learning through Demonstration and Eye Tracking. in *2017 IEEE International Conference on Robotics and Automation.* Institute of Electrical and Electronics Engineers, pp. 5921-5928, 2017 IEEE International Conference on Robotics and Automation, Singapore, Singapore, 29/05/17. https://doi.org/10.1109/ICRA.2017.7989697

Digital Object Identifier (DOI):
10.1109/ICRA.2017.7989697

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
2017 IEEE International Conference on Robotics and Automation

Physical Symbol Grounding and Instance Learning through Demonstration and Eye Tracking

Svetlin Penkov[1], Alejandro Bordallo[1] and Subramanian Ramamoorthy[1]

***Abstract*— It is natural for humans to work with abstract plans which are often an intuitive and concise way to represent a task. However, high level task descriptions contain symbols and concepts which need to be grounded within the environment if the plan is to be executed by an autonomous robot. The problem of learning the mapping between abstract plan symbols and their physical instances in the environment is known as the problem of physical symbol grounding. In this paper, we propose a framework for Grounding and Learning Instances through Demonstration and Eye tracking (GLIDE). We associate traces of task demonstration to a sequence of fixations which we call *fixation programs* and exploit their properties in order to perform physical symbol grounding. We formulate the problem as a probabilistic generative model and present an algorithm for computationally feasible inference over the proposed model. A key aspect of our work is that we estimate fixation locations within the environment which enables the appearance of symbol instances to be learnt. Instance learning is a crucial ability when the robot does not have any knowledge about the model or the appearance of the symbols referred to in the plan instructions. We have conducted human experiments and demonstrate that GLIDE successfully grounds plan symbols and learns the appearance of their instances, thus enabling robots to autonomously execute tasks in initially unknown environments.**

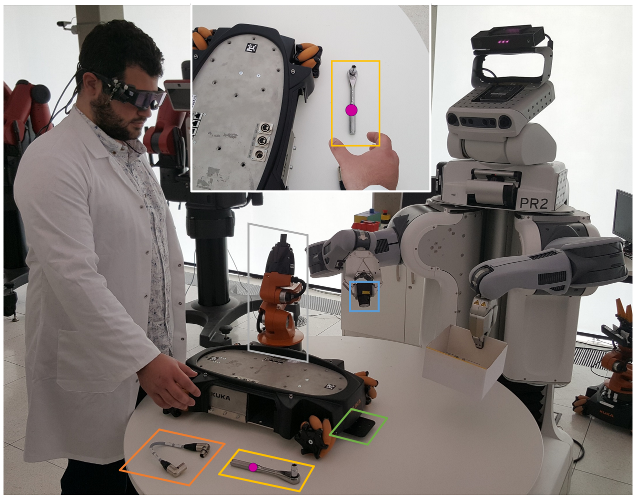


Fig. 1. Collaborative assembly settings demand understanding of contextualized information within cluttered and dynamic environments in order to understand human instructions and intents. Recognising task relevant objects and locations is also necessary in order to instantiate high level plans. As we demonstrate in this paper, eye tracking information enables symbol grounding and learning the appearance of symbol instances.

I. INTRODUCTION

Despite significant recent advances in the autonomous capabilities of humanoid robots, much remains to be done before robots are able to function effectively in complex human environments. This is especially the case when robots require understanding of contextualized information within cluttered and dynamic environments such as the collaborative assembly setup shown in Fig. 1. It is natural for humans to work with abstract plans for various tasks as they are often intuitive and concise. The ability to understand and autonomously execute high level instructions is crucial for the development of natural human robot interfaces (HRI). However, high level instructions contain symbols and concepts which need to be interpreted and grounded within the environment in order to enable plan execution. Work, in the field of HRI, related to interpreting abstract instructions (usually in the form of natural language) approaches the problem as grounding the unknown references within an instruction to a set of predefined actions and observed objects or locations in the environment [1], [2]. The problem of learning the mapping between abstract symbols and their physical instances in the environment, also known as the problem of *physical symbol grounding* [3], connects the idea of situated robotics [4] to the more general problem of symbol grounding [5] which is one of the core challenges in artificial intelligence [6].

Importantly, it is often the case that a robot may not have any knowledge about the model or the appearance of a symbol instance and so assuming a fully observable environment imposes considerable constraints on the robustness, scalability and applicability of the existing methods. In order to relax the observability assumption, we introduce GLIDE (Grounding and Learning Instances through Demonstration and Eye tracking) - a framework for simultaneously grounding symbols to their instances in the environment and learning their appearance. GLIDE enables robots to instantiate a high level plan within an initially unknown environment and then complete the task autonomously.

Our framework is based on learning from a small number of demonstrations by a person wearing eye tracking glasses. Considering the exponential development of technologies such as ubiquitous computing and the Internet of Things, being able to naturally communicate with a robot while wearing an eye tracking device seems an extremely plausible future. We exploit the fact that fixations are highly dependent on the task and provide information about the location of task related items. We describe a methodology for recording 3D fixation coordinates in the environment and call the sequence of fixations during task execution a *fixation program*. We

[1] School of Informatics, The University of Edinburgh, EH8 9AB, UK

formulate the problem of mapping fixation programs to high level task plans as probabilistic inference and demonstrate how this mapping can be used to ground plan symbols and learn appearance distributions of symbol instances in the environment. We recognise probabilistic programming as a tool well suited for the problem and so use Anglican [7] for the implementation. We tested GLIDE on experimental data from human demonstrations and confirm that it successfully performs physical symbol grounding. Overall our main contributions as presented in this paper are:

- Methodology for recording 3D fixations within the environment based on visual SLAM.
- An inference algorithm exploiting the properties of fixation programs in order to ground symbols and localise their instances in the environment.
- An algorithm for learning the appearance of symbol instances when no previous knowledge is present.

II. BACKGROUND

A. Symbol Grounding

The problem of grounding symbols to their meaning was introduced as the "Chinese Room" experiment [8] and later formally defined as the symbol grounding problem by Harnard [5]. A symbol is any object that is part of a symbol system and symbols are arbitrary in their shape. A symbol system is a set of symbols and syntactic rules for manipulating them on the basis of their shapes (not their meanings). Since robots are embedded and situated agents, our interpretation of symbol meaning follows the *physical symbol grounding* paradigm proposed by [3] and is a "functional relation between a form and a referent". In other words, the meaning of a symbol is its relation to the physical entity in the environment which the symbol is referring to.

Previous work related to symbol grounding and robotics focuses on human-robot interaction scenarios where natural language is used to provide commands to the robot [1], [2]. The syntactic constituents of the utterance are grounded to entities of interest in the environment - objects, locations, trajectories, actions, events. Interestingly, the utterance can also be semantically parsed to a program expressed in a robot control language [9] which the robot can execute. Multimodal models fusing gestures and language have also been proposed [10], [11]. However, a common assumption is that all entities of interest in the environment are observable. GLIDE aims to relax this constrain by enabling simultaneous grounding of the plan symbols and learning of the appearance of their instances. Importantly, the proposed framework utilises a high level planning language, instead of natural language, for the representation of plans.

B. Eye Tracking

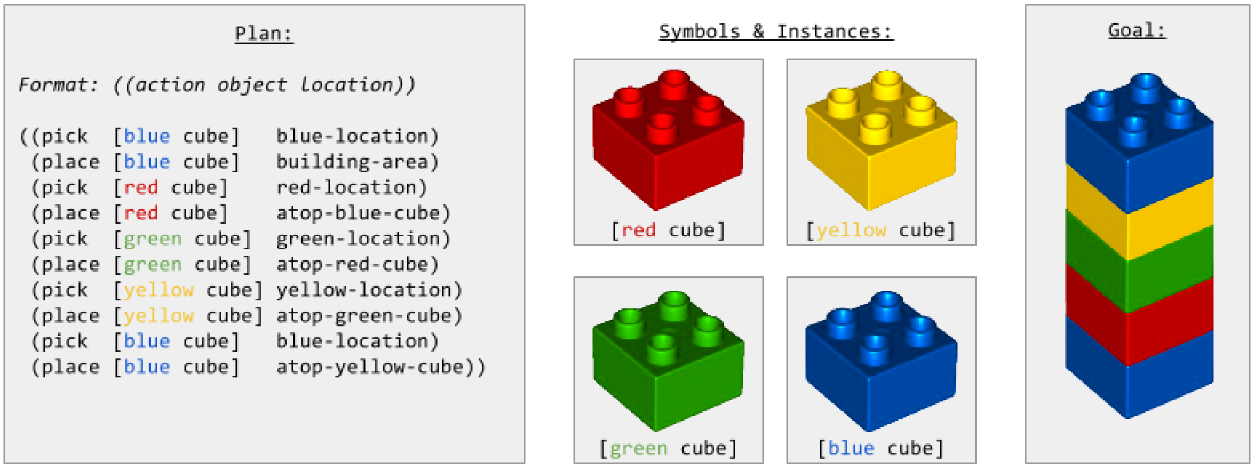
Oculomotor control has been studied during the performance of various tasks such as sandwich making, tea making, driving, playing table tennis, playing cricket and others (see [12] for an extensive review). The results from those studies show that eye movements during task execution are task specific and are barely affected by low level features in the environment [13], [14]. This supports the active vision paradigm according to which the vision system actively seeks information that is relevant to the current cognitive activity [15]. For example, Land et al. conducted a study [16] in which participants were asked to prepare a cup of tea in a regular kitchen environment while their eye movements were recorded. The majority of fixations were on objects related to the task despite the complexity of the environment and the free motion of the participants.

Eye tracking glasses (ETG) provide fixation locations as 2D points within the image from a first person point of view camera mounted on the device and facing the scene. Paletta et al. demonstrated that fixations can be projected onto a precomputed 3D map, by utilising SIFT features in the scene images [17], while Pfeiffer et al. rely on artificial fiducial markers in order to estimate 3D fixation locations [18]. Our approach utilises visual SLAM in order to reduce the constrains on the scene.

The predominant use of eye tracking in the field of robotics is in human-robot interaction settings, where gaze information enables the recognition of human behaviour [19] and the execution of anticipatory actions [20]. However, we focus on the question of how eye tracking can guide instruction grounding and perception. In this line of thought, Papadopoulos et al. treat fixations within an image as a noisy supervisory signal for the training of visual object class detectors [21], however they do not reason about fixations during the execution of multi-action tasks.

III. PROBLEM DESCRIPTION

Initially, a task such as building the tower of cubes shown in Fig. 2 (right) is demonstrated by a person wearing eye tracking glasses. We make no assumptions about the availability of prior knowledge related to the environment, however the task to be demonstrated $\mathcal{T}$ is predetermined. Additionally, the robot has access to a dictionary with action primitives $\mathcal{A}$ which the person can execute in an arbitrary sequence in order to solve the task. As illustrated in Fig. 2 (left), we represent plans for solving the task as a sequence of `(action object location)` tuples where `action` is symbol corresponding to an action from $\mathcal{A}$, `object` is a symbol referring to a physical object in the environment and `location` is a symbol referring to a physical location in the environment, sometimes dependent on the previous action. Even though `action` is a primitive action and known in advance, it can be executed by the robot only after the `object` and `location` symbols are grounded to their instances within the environment as shown in Fig. 2 (middle). Given that a person wears an ETG device and demonstrates how to execute the plan, the problem we address is how eye tracking information can be used to ground plan symbols to their instances in the environment and learn to recognise those in order to perform the demonstrated task autonomously.



```
Format: ((action object location))

((pick  [blue cube]   blue-location)
 (place [blue cube]   building-area)
 (pick  [red cube]    red-location)
 (place [red cube]    atop-blue-cube)
 (pick  [green cube]  green-location)
 (place [green cube]  atop-red-cube)
 (pick  [yellow cube] yellow-location)
 (place [yellow cube] atop-green-cube)
 (pick  [blue cube]   blue-location)
 (place [blue cube]   atop-yellow-cube))
```

Fig. 2. The representation of a plan **(left)** which solves an example task with the goal of building a tower with five colour cubes in a particular order **(right)**. All plan symbols need to be grounded to their physical instances **(middle)** in order to enable autonomous execution of the task.

IV. METHODOLOGY

We split the problem of simultaneous symbol grounding and instance learning in three parts. First, we describe how to estimate 3D fixation locations within the environment in order to map fixations to physical locations in the environment. Secondly, we show how to localise the instance of a plan symbol by exploiting the properties of a fixation program. Lastly, once the location of an instance is known, we describe how to learn its appearance in order to enable automatic recognition.

A. 3D Eye Tracking

Mobile eye trackers provide fixation information in pixel coordinates corresponding to locations in the image of a first person view camera. Relying only on that information requires all items of interest in the environment to be detected from the image in order to determine which one is fixated. However, in a collaborative assembly scenario, such as the one shown in Fig. 1, there are several sensors which can be used - the robot has an RGBD sensors, multiple cameras and laser scanners. Additionally, the environment can also be highly sensorised. In order to take advantage of multiple sensors, instead of a single first person view camera, we estimate the 3D fixation locations, enabling the projection of fixations in the frame of any sensor in the environment.

In order to achieve that, we calibrate the first person view camera and utilise the ORBSLAM algorithm [22], which is a mono camera SLAM algorithm, in order to estimate the 6D pose of the eye tracking glasses in real time. Since a single camera is used, the obtained measurements are correct only up to a scale factor. Therefore we have developed a calibration procedure which relies on detecting several APRIL tags [23] initially present in the scene. First, the physical distance d_{calib} between two different view points is estimated using the calibration parameters of the camera and the tags. Then, the distance d_{vslam} is calculated from the estimated poses by ORBSLAM and the scale is calculated as d_{calib}/d_{vslam}. Additionally, the transformation between the world frame and the origin of the ORBSLAM frame is also also computed during the calibration procedure. Once the 6D pose of the glasses is calculated within the world frame, the fixation locations are projected from the first person view camera image to the 3D environment by ray casting and finding the intersection with a 3D model of the environment. As a result, fixations can be represented as 3D locations in the environment instead of just pixel coordinates. We are interested in scenarios where table top manipulation is required, so we project fixations to the plane of the table top resulting in 2D points with physical coordinates. It is also possible to intersect the ray with a point cloud of the environment [17].

Utilising a visual mono SLAM algorithm for pose estimation of the eye tracking glasses reduces the number of assumptions about the scene, however it puts demands on the hardware. Throughout our work we use the SMI ETG 1 device which provides fixations information at a rate of $60Hz$ and the first person view camera has a frame rate of $30FPS$. Initial experiments revealed that this frame rate is insufficient for the ORBSLAM algorithm since people move their heads relatively fast, adding significant blur to the images. Therefore, we have attached an extra camera to the eye tracking glasses which provides $120FPS$ and enables robust head tracking. The transformation between the frame of the eye tracking glasses camera and the high frame rate one is also estimated during the calibration procedure.

B. Model Definition

In order to solve the problem of physical symbol grounding we define a generative probabilistic model shown in Fig. 3. The task $\mathcal{T}$ and the fixation program $F_1 \dots F_T$ are observed, where $F_i \in \mathbb{R}^2$ and T is the total number of observed fixations. P encodes any valid plan which successfully completes the task. Following the idea that "task and context determine where you look" [15], the action $A_i \in \mathcal{A}$ induces a set of fixations $F_{s_i} \dots F_{l_i}$ on a certain item of interest related to the action, where $s_1 = 1$ and

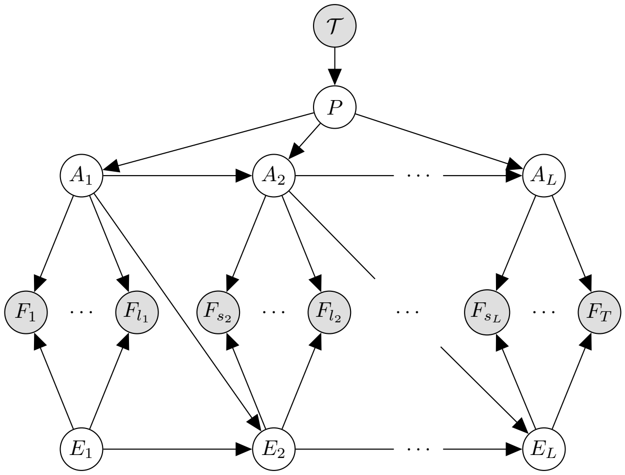


Fig. 3. The proposed probabilistic model for physical symbol grounding which is based on the idea that "task and context determine where you look" [15].

$l_L = T$. In the context of planning, an item of interest could be any of the symbols in the plan. For example, given the plan in Fig. 2 (left), an item of interest might be either a particular cube or a location such as the `building-area`. In general, $l_i + 1 - s_{i+1} > 0$ because fixations are observed, even though sparse, during the transition from one item of interest to another. Those fixations are not represented in the graphical model in order to avoid clutter. Additionally, the set of fixations $F_{s_i} \dots F_{l_i}$ also depends on the actual physical location of the item of interest. The main assumption is that symbol instances cannot be recognised prior to learning, therefore we model the belief about the position of item m as a normal distribution $\mathcal{N}_m(\mu_m, \Sigma_m)$ over possible locations in the environment. Since fixations are projected to the table top plane $\mu_m \in \mathbb{R}^2$ and Σ_m is a 2×2 covariance matrix. μ_m and Σ_m are latent variables which we are interested in inferring in order to consequently learn the instance of symbol m. The set of tuples $\{(\mu_i, \Sigma_i) : i = 1 \dots M\}$ which encodes the state of the environment is denoted as E, where M is the number of items of interest, or equivalently the number of symbol instances. We assume that the changes in the environment from state E_i to state $E_i + 1$ can be caused only by the action A_i.

C. Inference

Assuming a uniform prior over the possible locations of each item and that a task can be uniformly chosen from a set of predefined ones, we are interested in solving the following inference problem

$$p(E_{1:L}|F_{1:T}, \mathcal{T}) \propto \\ \propto \sum_{P, A_{1:L}, s_{1:L}, l_{1:L}} p(F_{1:T}|A_{1:L}, E_{1:L}) p(A_{1:L}|P) p(P|\mathcal{T}) \tag{1}$$

where we use the notation $X_{a:b}$ to express the sequence $X_a, X_{a+1}, \dots, X_{b-1}, X_b$. The key insight is that fixation programs encode information about why a certain point is fixated (what action is performed) and where the fixation is located (where the action item of interest is). Reasoning about those two aspects can be performed independently as

$$p(F_{1:T}|A_{1:L}, E_{1:L}) = p(F_{1:T}|A_{1:L}) p(F_{1:T}|E_{1:L}) \tag{2}$$

By exploiting the properties of fixation programs, we split the inference problem in 4 parts in order to obtain a computationally feasible solution.

Algorithm 1: Environment Inference

```
Input: $\mathcal{T}$, $\mathcal{A}$, $\mathcal{P}$, $\lambda_m$, $\lambda_s$, $D_{max}$
Data: $F_{1:T}$
Output: $E = \{(\mu_1, \Sigma_1), (\mu_2, \Sigma_2), \dots, (\mu_M, \Sigma_M)\}$
// Find all possible plans for the task
$P_\mathcal{T} = \mathcal{P}(\mathcal{T})$
$samples \leftarrow []$
for $n$ from $1$ to $N_{samples}$ do
    // Step 1: Generate actions
    $P^n \leftarrow$ Sample $UniformCategorical(P_\mathcal{T})$
    $L^n \leftarrow GetLength(P^n)$
    $A^n_{1:L^n} \leftarrow GetActions(P^n)$
    // Step 2: Generate fixation program segments
    $\alpha^n \leftarrow 2L^n$
    $[s^n_1, l^n_1, \dots s^n_L, l^n_L] = T *$ Sample $Dir(\alpha^n)$
    // Step 3: Evaluate fixation program likelihood
    $w^n \leftarrow 1.0$
    for $i$ from $1$ to $T - 1$ do
        if $BothDuringAction(F_i, F_{i+1})$ then
            $d_m \leftarrow ||F_i - F_{i+1}||$
            $w^n \leftarrow w^n * \lambda_m \exp(-\lambda_m d_m)$
        if $BothDuringTransition(F_i, F_{i+1})$ then
            $d_s \leftarrow D_{max} - ||F_i - F_{i+1}||$
            $w^n \leftarrow w^n * \lambda_s \exp(-\lambda_s d_s)$
    // Store sample
    $sample \leftarrow \{w^n : [s^n_1, l^n_1, \dots s^n_L, l^n_L]\}$
    Append $sample$ to $samples$
$[\hat{s}_1, \hat{l}_1, \dots \hat{s}_L, \hat{l}_L] = WeightedAverage(samples)$
// Step 4: Estimate item of interest locations
$E \leftarrow []$
for $m$ from $1$ to $M$ do
    $(\mu_m, \Sigma_m) = FitNormal(F_{\hat{s}_m : \hat{l}_m})$
    Append $(\mu_m, \Sigma_m)$ to $E$
```

1) Generating actions: The term $p(P|\mathcal{T})$ encodes the probability of a plan given the demonstrated task and is computed by a high level planner $\mathcal{P}$. The planner is assumed, by utilising the dictionary of primitive actions $\mathcal{A}$, to find the set of all plans $P_\mathcal{T}$ that successfully achieve the task. $p(P|\mathcal{T})$ is defined as a uniform categorical distribution over $P_\mathcal{T}$. By biasing the categorical distribution, it is possible to represent any preferences with respect to a plan, for example shorter plans are selected with higher probability. $p(A_{1:L}|P)$ is similar to an indicator function and assigns a probability of 1 to the sequence of actions $A_{1:L}$ which is defined by plan P and 0 to any other sequence.

2) Generating fixation program segments: The main difficulty in expressing the likelihood of the observed fixations given the actions as shown in (2) is the fact that there are T fixations and L actions. The fixations induced by action A_i are $F_{s_i:l_i}$, while the fixations made during a transition from action A_i to A_j are given by $F_{l_i+1:s_j-1}$. If there are L actions then there are $L-1$ additional transitions resulting in $2L-1$ sets of fixations in total. Thus, the term $p(F_{1:T}|A_{1:L})$ in (2) can be rewritten as

$$p(F_{1:T}|A_{1:L}) = p(F_{1:l_1}, F_{l_1+1:s_2-1}, F_{s_2:l_2} \dots F_{s_L:l_L}|A_{1:L})$$

Similar to the stick breaking analogy for sampling from a Dirichlet distribution [24], the sequence $s_1, l_1, s_2, l_2, \dots s_L, l_L$ can be viewed as points where the fixation program is split in $2L-1$ segments with total length equal to T. Therefore, $s_1, l_1, s_2, l_2, \dots s_L, l_L$, are sampled from a $(2L-1)$-dimensional symmetric Dirichlet distribution with concentration parameter α and normalised such that $s_1 = 1$ and $l_L = T$. Since there are no zero length segments in the fixation program α should be greater than 1. Empirical tests showed that setting $\alpha = 2L$ yields a stable heuristics. It should be noted that sampling the fixation program segments is equivalent to sampling the structure of the graphical model in Fig. 3.

3) Evaluating fixation program likelihood: Fixations on items of interest are clustered on the item, while fixations made during transition are sparse and have relatively large distance between each other. Therefore, the likelihood $p(F_{1:T}|A_{1:L})$ is modelled as

$$p(F_{1:T}|A_{1:L}) = \prod_{i}^{T-1} L_{item}(F_i, F_{i+1}) L_{trans}(F_i, F_{i+1}) \quad (3)$$

The likelihood of two consecutive fixations F_i and F_j during an action A_k is defined as an exponential distribution over the distance between them $d_m = ||F_i - F_j||$, thus

$$L_{item}(F_i, F_j) = \\ = \begin{cases} \lambda_m e^{-\lambda_m d_m} & \text{if } F_i, F_j \in [s_k, l_k] \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

which means that fixations closer together are more likely. λ_m is the mean distance between consecutive fixation during an action and is learnt from labelled data. The likelihood of two consecutive fixations F_i and F_j during a transition from A_k to A_{k+1} is defined as an exponential distribution over the $d_s = D_{max} - ||F_i - F_j||$, where D_{max} is the maximum possible distance between F_i and F_j, and so

$$L_{trans}(F_i, F_j) = \\ = \begin{cases} \lambda_s e^{-\lambda_s d_s} & \text{if } F_i, F_j \in [l_k + 1, s_{k+1} - 1] \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

In this case more distant fixations are more likely and λ_s is the mean distance between consecutive fixation during a transition and is also learnt from labelled data. D_{max} can be learnt from data as well, however we have set it simply to the diameter of the circular table on which the experiments were conducted. We perform importance sampling by using the described likelihood function in order to obtain the estimates $\hat{s}_1, \hat{l}_1, \hat{s}_2, \hat{l}_2, \dots \hat{s}_L, \hat{l}_L$.

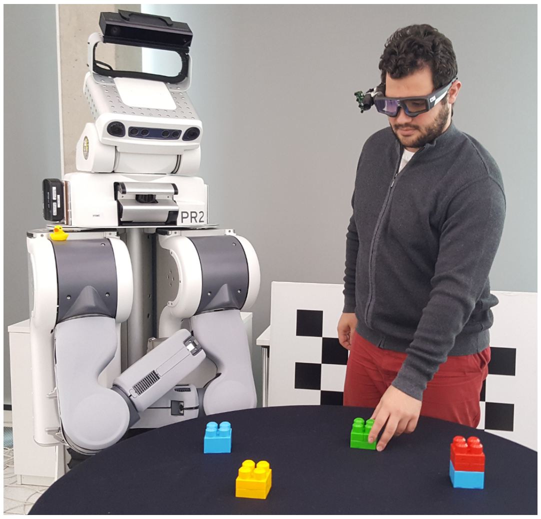


Fig. 4. The experimental setup used for recording demonstrations of building a tower with five colour cubes. During the demonstration the person wears eye tracking glasses which we have modified to enable visual mono SLAM based localisation.

4) Estimating item of interest locations: Once the partitioning of the fixation program is estimated, inferring the location of each item of interest is performed through maximum likelihood estimation. For each item of interest i we fit $\mathcal{N}_i(\mu_i, \Sigma_i)$ to the fixations segment $F_{\hat{s}_i:\hat{l}_i:}$ corresponding to the action A_i. Due to the noisy nature of the eye tracking signal we constrain Σ_i to be a diagonal matrix and so avoiding potential problems with overfitting to the fixation clusters.

The pseudo code in Alg. 1, which is optimised for clarity rather than efficiency, summarises the proposed algorithm. We have implemented it with the Anglican probabilistic programming language [7].

D. Instance learning

Once the symbol instances are estimated, the instance learning aim is to learn their appearance in order to enable automatic recognition. As suggested by [21], grounded fixations can be used as a noisy supervisory signal for any type of classifier. Additionally, since the estimated locations can be projected in the frame of any sensor it is possible to train multimodal classifiers. We describe a simple vision based algorithm for instance learning as an initial step towards the development of more sophisticated systems. Given a set of fixations $F_{s_i:l_i}$ which correspond to an item of interest, it is projected onto the image of a camera viewing the scene resulting in $f_{s_i:l_i}$. For each projected fixation an image crop is made centred at the fixation with size proportional to the variance of $\mathcal{N}_i$. The difference between every two crops is computed in order to estimate the colour of the background. After that each crop is resized to enclose the largest foreground object it contains. The resized crops are

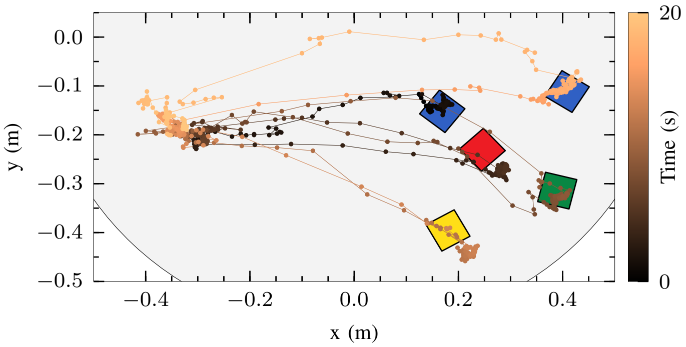


Fig. 5. A typical fixation program recorded during a demonstration of how to build a tower of 5 colour cubes in the sequence of blue, red, green, yellow, blue (from bottom to top).

used to compute the distribution over the colour and size of the symbol instance. This information is sufficient to detect the object in simple environments such as the table top setup. It should be noted that different sensors will lead to different appearance distributions and so a more general representation can be learnt by taking multiple sensors into account.

E. Experimental setup

We have conducted human experiments where the PR2 robot is shown how to build a tower of five colour cubes such as the one in Fig. 2 (right). Each of the 5 participants demonstrated the task 10 times, working on a table top, while wearing eye tracking glasses which we have modified as can be seen in Fig. 4. We recorded eye tracking information as well as video feeds from multiple cameras together with the poses estimated by ORBSLAM. One of the cameras, which is fixed on the ceiling, is used for estimating ground truth locations of the cubes.

V. RESULTS

A. 3D Eye Tracking

The first step in the evaluation of GLIDE is to analyse the results obtained by the proposed 3D eye tracking methodology. Correct pose estimation of the eye tracking glasses is crucial for the projection of fixations on the environment, therefore we have tested multiple visual SLAM algorithms on the table top setup which we are interested in. ORBSLAM relies on image feature points and does not assume that they belong to a single plane. Thus, it is able to use feature points detected both on the table and in the environment which we found to be crucial. The only problems which we experienced were with people leaning over the table, looking at it from closer and so limiting the number of visible feature points. However, the issue was easily resolved by tilting the camera up slightly and recalibrating its transformation.

A typical fixation program recorded during a demonstration of how to build a tower of 5 colour cubes in the sequence of blue, red, green, yellow, blue (from bottom to top) is shown in Fig. 5. Firstly, it can be noticed that fixations are indeed clustered on items of interest for each action and are sparse in the transition stages, which is a key assumption in the proposed inference algorithm. We used one demonstration from each participant to estimate the mean distances between fixation for the two cases $\lambda_m = 0.81cm$ and $\lambda_s = 6.29cm$ respectively. As expected, λ_s is an order of magnitude greater than λ_m. This difference is crucial as it provides valuable information in order to align the plan with the observed fixation program. Furthermore, cluster locations are not directly on the cubes, but often at the edge or even slightly aside. On one hand, this can be explained by the approximation of projecting fixations on the table top, instead of the top side of each cube. This distortion effect can be easily noticed in the building area where each consequent cube violates the planar assumption stronger than the previous one and the fixations form a diagonal cluster. The person is usually positioned at approximately $(0.0, -0.6)$ which matches the angle of the diagonal cluster. On the other hand, people are known to fixate on task critical locations such as grasp points [25]. Another interesting feature is that trajectories from the first block to the building area follow almost straight line paths, while the last one is noticeably curved. This pattern is present in most of the fixation programs which we recorded and we attribute it to the fact that people change their focal plane during the transition, however we are not able to detect that and simply project the fixation onto the table top. This can be avoided by monitoring the 3D optical axis of each eye and find the

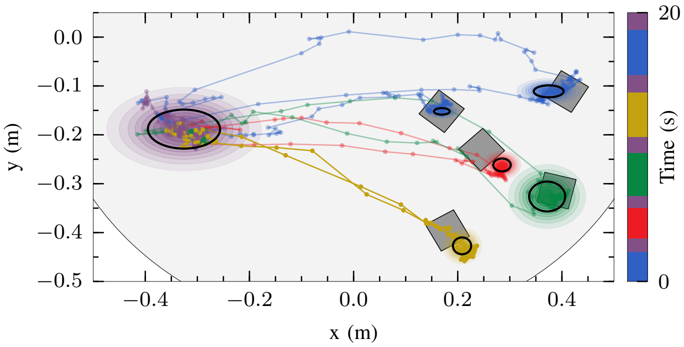


Fig. 6. The results from running the proposed inference algorithm on the fixation program in Fig. 5 in order to determine the location of each symbol instance in the environment. The ellipses represent the inferred $\mathcal{N}_m(\mu_m, \Sigma_m)$ for each item of interest. The black ellipses correspond to a single standard deviation. The purple ellipse represents the location of the building area obtained by averaging the 5 building steps. The transition fixations are not explicitly visualised.

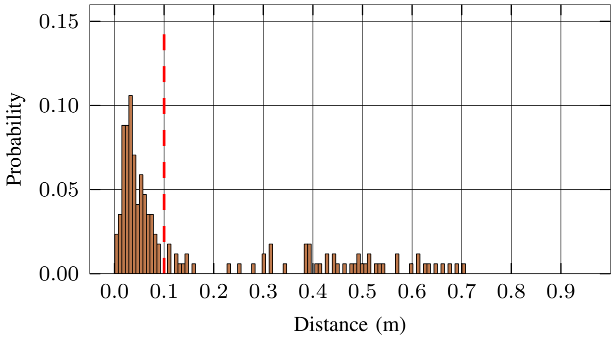


Fig. 7. A probability mass distribution over the error in localising each of the cubes in each of the recorded demonstrations. The error is calculated as the distance between the ground truth locations and the inferred locations from the fixation programs.

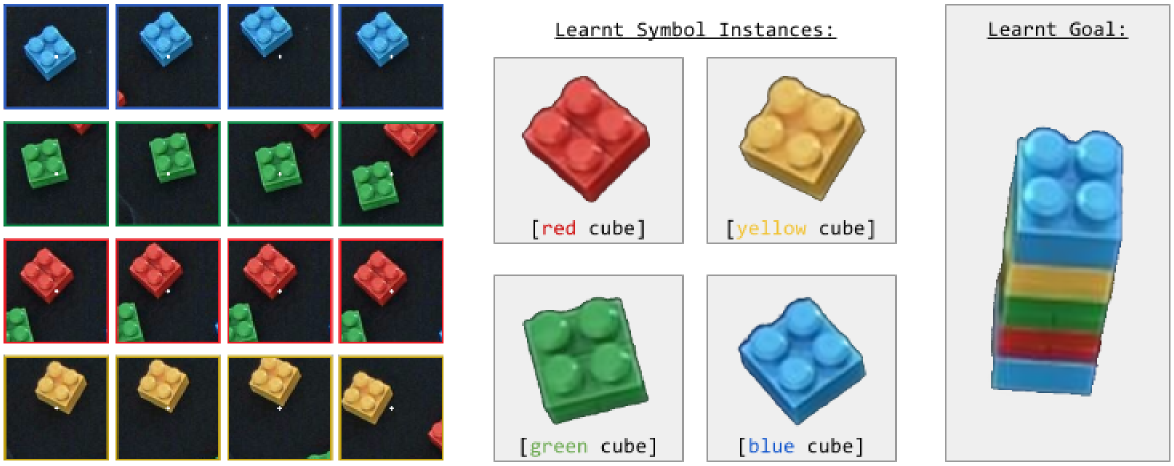


Fig. 8. Images of symbol instances extracted by utilising the results from performing inference over the recorded fixation program **(left)**. Each of the crops is centred at a fixation belonging to the corresponding action segment. Those images are used to ground the plan symbols to their physical instances and learn their appearance **(middle)**. The visualised instances are the mean of the learnt appearance distributions. The approach can also be used to learn the appearance of the task goal **(right)**.

intersection between them in order to truly estimate a 3D fixation. The SMI ETG 1 device provides such information however we found it to be extremely noisy. Therefore, we rely on the point of regard within the first person view camera image which is less noisy, but abrupt changes with large magnitude are often observed. One of the preprocessing steps that we employed is to remove any fixations which are out of the table surface.

B. *Localisation of Symbol Instances*

Next we proceed with the evaluation of the proposed inference algorithm. The inferred locations from the fixation program in Fig. 5 are shown in Fig. 6. Each $\mathcal{N}_m(\mu_m, \Sigma_m)$ for each item m is visualised as an ellipse with the corresponding colour. The purple ellipse represents the estimated location for the building area which was calculated by averaging the locations of the five building steps. It can be seen that the fixation program was segmented correctly and each symbol instance was successfully localised. The building area has the greatest variance which is expected as there are 5 actions depending on it. In order to evaluate the performance of the inference algorithm we have plotted in Fig. 7 the distribution of localisation error as a histogram by analysing all recorded demonstrations. The error is calculated independently for each cube and it is defined as the distance between the ground truth location estimated by a top view camera and the inferred location from the fixation program. 73% of the cubes are successfully localised with an error of less than $10cm$ which is comparable to the cube size of $6.5cm$. Manual inspection of erroneous cases revealed that wrong localisation is predominantly caused by noisy eye tracking data that we have not filtered out. However, since we segment the fixation program and the beginning and the end are fixed, usually only a small number of item of interest locations are affected by the noise.

C. *Instance Learning*

The last step in the evaluation of the proposed framework is to examine the performance of the vision based instance learning algorithm. Given that the location of a symbol instance is inferred, we can take image crops around the corresponding fixations as shown in Fig. 8 (left). Each crop contains the item of interest being fixated, however, additional items are also partially visible due to the cluttered table top. Nevertheless, the proposed instance learning algorithm manages to filter the extra items out and calculates an appearance distribution over the size, colour and pixel values of each symbol instance. The mean value of each symbol instance is visualised in Fig. 8 (middle). Manual visual inspection of the learnt appearance models for each cube throughout all demonstrations revealed that 71% of the instances were correctly learnt. Furthermore, the same approach can be used to learn the appearance of the task goal as demonstrated in Fig. 8 (right). There is an interesting trade off between the number of demonstrations, number of sensors and generalisation capabilities of the learnt appearance models. The ones shown in Fig. 8 (middle) are learnt from a single static camera from a single demonstration. While they look surprisingly similar to the real cubes, they have overfit the appearance of the symbol instance and slight changes in the point of view or lighting conditions will render them invalid. Therefore, it is an interesting problem on its own how to combine multiple sensors, possibly moving, with a number of demonstrations in order to arrive at a more general appearance model of the given symbol instance.

VI. DISCUSSION

In this work we have presented how the interpretation of high level instructions together with 3D eye tracking data can actively guide perception and enable robots to instantiate symbolic plans without prior knowledge about the symbol instances present in the environment. This type of situations are inevitable if we are to deploy autonomous robots in actual human environments so that they can improve our daily lives.

We have demonstrated that fixation programs in the context of plan execution provide not only information about where people look at, but also why they look there. This property has enabled us to develop a generative probabilistic model for fixation programs together with an efficient inference algorithm. The proposed instance learning algorithm attempts to answer the last question about what people see when they look at a particular location. Thus, GLIDE is able to interpret unknown symbol references present in single instructions or entire plans by mapping them to 1) a location within the environment and 2) a corresponding physical appearance. We demonstrated the capabilities of GLIDE on experimental data where 73% of the symbols were correctly localised and the physical appearance of 71% of the symbols was successfully learnt.

One of the main limitations of the proposed inference algorithm is that it will fail to segment the fixation program properly if informative actions such as searching are executed by the demonstrator. One way to resolve this issue is to add a prior over the number of fixation clusters in the fixation program. However, the planner should also be able to predict such actions.

In fact, the planner adds another constraint which should be discussed. We assume that the planner is able to generate all possible plans for accomplishing the task. This is possible for tasks with low branching factor where the order is important. If we consider a task which has a high branching factor and the order does not matter then the number of plans grows exponentially rendering an exhaustive search approach infeasible. A potential solution of this issue is to keep a recursive estimate of the most probable plan currently being executed. This, however, will not work if no knowledge about the environment is present. Therefore, it is left for future work to integrate natural language instructions in GLIDE in order to learn the symbolic plan and then instantiate it in the environment.

VII. Conclusion

In this paper, we introduced GLIDE - a framework for simultaneous Grounding and Learning Instances through Demonstration and Eye tracking. We demonstrated that it successfully manages to ground symbols and learn their the appearance by applying it to experimental data. The key insight was the definition of fixation programs as associated traces of demonstrations and fixation sequences. This enabled us to explore how eye tracking can guide the instantiation of high level plans as well as perception and environment understanding. Those are key capabilities necessary for the successful deployment of robots in human environments.

In conclusion, GLIDE is a tool enabling robots to deal with high level instructions without assuming any prior knowledge about the symbol instances present in the environment.

References

[1] S. Tellex, P. Thaker, J. Joseph, and N. Roy, "Learning perceptually grounded word meanings from unaligned parallel data," *Machine Learning Journal (Special Issue on Learning Semantics in Machine Learning)*, vol. 94, no. 2, pp. 151–167, 2014.

[2] T. Kollar, S. Tellex, M. R. Walter, A. Huang, A. Bachrach, S. Hemachandra, E. Brunskill, A. Banerjee, D. Roy, S. Teller, *et al.*, "Generalized grounding graphs: A probabilistic framework for understanding grounded language," *Journal of Artificial Intelligence Research*, 2013.

[3] P. Vogt, "The physical symbol grounding problem," *Cognitive Systems Research*, vol. 3, no. 3, pp. 429–457, 2002.

[4] R. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, pp. 139–159, 1991.

[5] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 335 – 346, 1990.

[6] M. Taddeo and L. Floridi, "Solving the symbol grounding problem: a critical review of fifteen years of research," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 17, no. 4, pp. 419–445, 2005.

[7] F. Wood, J. W. van de Meent, and V. Mansinghka, "A new approach to probabilistic programming inference," in *Proceedings of the 17th International conference on Artificial Intelligence and Statistics*, 2014, pp. 1024–1032.

[8] J. R. Searle, "Minds, brains, and programs," *Behavioral and Brain Sciences*, vol. 3, pp. 417–424, 9 1980.

[9] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Experimental Robotics*. Springer, 2013, pp. 403–415.

[10] D. Whitney, M. Eldon, J. Oberlin, and S. Tellex, "Interpreting multimodal referring expressions in real time," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3331–3338.

[11] C. Matuszek, L. Bo, L. Zettlemoyer, and D. Fox, "Learning from unscripted deictic gesture and language for human-robot interactions." in *AAAI*, 2014, pp. 2556–2563.

[12] M. F. Land, "Eye movements and the control of actions in everyday life." *Progress in retinal and eye research*, vol. 25, no. 3, May 2006.

[13] M. Hayhoe and A. Shrivastava, "Visual memory and motor planning in a natural task," *Journal of Vision*, pp. 49–63, 2003.

[14] M. Hayhoe and D. Ballard, "Eye movements in natural behavior." *Trends in cognitive sciences*, vol. 9, no. 4, pp. 188–94, 2005.

[15] C. A. Rothkopf, D. H. Ballard, and M. M. Hayhoe, "Task and context determine where you look." *Journal of vision*, vol. 7, 2007.

[16] M. Land, N. Mennie, and J. Rusted, "The roles of vision and eye movements in the control of activities of daily living," *Perception*, vol. 28, no. 11, pp. 1311–1328, 1999.

[17] L. Paletta, K. Santner, G. Fritz, H. Mayer, and J. Schrammel, "3d attention: measurement of visual saliency using eye tracking glasses," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 199–204.

[18] T. Pfeiffer, P. Renner, and N. Pfeiffer-Leßmann, "Eyesee3d 2.0: Model-based real-time analysis of mobile eye-tracking in static and dynamic three-dimensional scenes," in *Proceedings of the Ninth Biennial Symposium on Eye Tracking Research & Applications*. ACM, 2016, pp. 189–196.

[19] W. Yi and D. Ballard, "Recognizing behavor in hand-eye coordination patterns," *International Journal of Humanoid Robotics*, vol. 06, no. 03, pp. 337–359, 2009.

[20] C.-M. Huang and B. Mutlu, "Anticipatory robot control for efficient human-robot collaboration," in *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, 2016, pp. 83–90.

[21] D. Papadopoulos, A. Clarke, F. Keller, and V. Ferrari, "Training object class detectors from eye tracking data," in *European Conference on Computer Vision (ECCV)*, 2014.

[22] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[23] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3400–3407.

[24] Y. W. Teh, D. Görür, and Z. Ghahramani, "Stick-breaking construction for the indian buffet process." in *AISTATS*, 2007, pp. 556–563.

[25] R. S. Johansson, G. Westling, A. Bckstrm, and J. R. Flanagan, "Eyehand coordination in object manipulation," *The Journal of Neuroscience*, vol. 21, no. 17, pp. 6917–6932, 2001.