Behaviour graphs: A semantic-contextual representation for robot navigation

Joel Loo, David Hsu
Adacomp Lab, National University of Singapore

***Abstract*—Humans are able to use *navigational behaviours* such as *follow the corridor* or *turn right* to represent directions and spatial information. Inspired by this, we explore how to incorporate behaviours into robot representations. We propose *behaviour graphs*, a topological representation in which nodes represent coarse locations, and edges represent behaviours used to transit between nodes. These graphs contain semantic labels of places along with contextual information in the form of robot-specific behaviours, and do not depend on metric information. We observe that navigational behaviours can be naturally inferred from second-hand descriptions of the world that may be abstract or inaccurate - e.g. floor-plans, hand-drawn maps. Thus we propose to construct such graphs through 'map-reading', leveraging the abundance of second-hand maps of manmade environments. We assemble a navigation system that uses such inferred behaviour graphs, and test its navigation performance on a real robot.**

I. Introduction

How might we direct a friend to our new apartment inside a building they have never been in? We might easily imagine instructing them as such: "After the foyer turn left, go straight down the corridor, take a right turn, then go to the door at the end." This scenario highlights how humans can rely on *navigational behaviours* such as *corridor-following* ("go straight down the corridor") or *turning* ("take a right turn") to represent directions and spatial information. These behaviours can be concretely characterised as navigation policies that are capable of collision avoidance, while also exploiting *navigational affordances* in the environment - e.g. by recognising paths that can be followed, or turns that can be made. Inspired by this behavioural representation, we explore how behaviours can be encoded in robot representations, and consider the utility they bring. In particular, we build on recent work in visual navigation implementing navigational behaviours with neural networks [9, 10, 1] and consider how such behaviours can influence the design of robot representations for navigation.

Since navigational behaviours are discrete, symbolic actions, encoding them into robot representations can enable us to create *state-action* representations that not only capture possible states in the environment, but the action-induced transitions between them. Such representations effectively memoise the sequence of high-level actions needed to complete each navigation task, simplifying motion planning compared to traditional *state-only* representations that require actions to be computed on-the-fly. Also, such representations could capture paths as human-interpretable directions, since navigational behaviours are often trained to align with human-like behaviours.

Using navigational behaviours can reduce dependence on metric information. Most robot systems rely on detailed and accurate metric SLAM maps to find feasible paths and localise. Building accurate maps over large scales remains a challenge for SLAM [2]. Navigational behaviours can help to sidestep this issue in two ways. Firstly, they are often able to generate collision-avoiding actions from visual input, without needing geometric reconstructions of the environment. Secondly, they can implicitly capture structural information about the environment, much like how the directions in the above scenario capture the building's topology and layout. Behaviour-based representations could potentially contain sufficient structural information to enable localisation without metric information.

Most crucially, navigational behaviours are a form of contextual scene information that can be extracted not just from direct experience of the environment, but also from second-hand descriptions of it. For instance, humans can infer the behaviours needed to reach a goal in a novel environment from natural language directions or floor-plans, all before stepping foot into the environment. Robots that use navigational behaviours can capitalise on the wealth of second-hand descriptions of manmade environments - ranging from satellite maps to abstract maps like floor-plans - to construct representations for navigating in hitherto unseen areas.

To realise these possibilities and investigate the practicality of such behaviour-based representations, we consider 3 questions: (1) What form should a behaviour-based representation take? (2) How can we generate this representation? (3) How can this representation be employed for navigation tasks?

To address (1), we formulate a semantic-contextual *behaviour graph* representation that represents a set of key places in the environment, with directed edges indicating connectivity between them. Select nodes may contain semantic labels related to the location they represent, while the edges also capture contextual, robot-specific action affordance information. Specifically, each edge is labelled with the specific navigational behaviour required to transit between the end-point nodes. In our implementation, we draw the navigational behaviours from the behaviour set proposed in Ai et al. [1]. For (2), we propose a learnable 'map-reading' pipeline that can be trained to extract behaviour graphs from commonly available top-down 2D 'maps' of the environment. In particular, we consider both metrically accurate 'maps' like satellite maps, as well as abstract and inaccurate 'maps' such as floor-plans and hand-drawn maps. To answer (3), we address how to plan and localise using behaviour graphs. Planning to a goal node in the graph can be achieved with graph search, and we adapt Graph Localisation Networks [3] to localise with respect to the behaviour graph.

We broadly observe from our experiments that behaviour graphs can be extracted from a variety of different 'map' types with reasonable accuracy. More importantly, we find that these noisy behaviour graphs inferred with our 'map-reading' can be employed for effective navigation on a real robot system.

II. Related Work

A. Navigational behaviours

Recent visual navigation methods have built neural networks capable of steering robots towards a specified goal using only visual input [11]. Such works implicitly encode the environment in their weights, hindering generalisation to novel areas. Sorokin et al. [10] instead build *navigational behaviours*, capturing semantically meaningful actions like sidewalk- or path-following that can generalise across environments. However, a single behaviour can only reach a limited subset of goals in the environment - to cover most possible goals, a *set* of behaviours is needed. In this vein, Sepulveda et al. [9], Codevilla et al. [4], Ai et al. [1] propose various sets of navigational behaviours for reaching goals in indoor and outdoor environments. While many works tackle the *implementation* of navigational behaviours, *encoding information* in representations with behaviours is less explored.

B. Robot representations

Modern robot representations are often explored in the context of SLAM, and usually (1) encode environment geometry and appearance for localisation and motion planning, and are (2) built from direct experience or observations of the environment [2]. Recent work recognises the need to also explicitly represent semantic information for decision-making. Grinvald et al. [5] build volumetric maps that separate structures from objects in the scene. Hughes et al. [6] organise semantic information into a hierarchical, topological structure that is extracted from an underlying SLAM map. However, these are built on metric SLAM maps which can be tricky to scale to large (kilometre-scale) areas. They also do not interpret the scene in the context of the robot agent - e.g. by representing the robot's action affordances in the scene. In view of this, this work aims to reduce reliance on metric SLAM and encode navigational affordances, by incorporating navigational behaviours in robot representations. Our work shares heritage with Kuipers [7], Sepulveda et al. [9], Chen et al. [3], all of whom propose various topological representations that explicitly represent navigational behaviours. In contrast with the above works, we observe that navigational behaviours can be inferred from abstract, second-hand 'maps' of the environment like floor-plans, and propose a pathway to construct our representations from such 'maps' as an alternative to collecting direct experience in the target environment.

III. Behaviour Graph Design

A behaviour graph is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node $v \in \mathcal{V}$ represents a coarse location in the environment, and is either a *destination* or *changepoint* node. Destination nodes are user-specified locations that are potential goals for the robot. These nodes may be tagged with semantic labels identifying the nature of the destination, e.g. 'kitchen' or 'foyer'. Changepoint nodes are locations at which the robot is afforded the chance to switch between navigational behaviours. We note that while nodes indicate coarse locations, no metric coordinates or information are associated with each node. Instead our system localises itself within the graph by matching the observed layout of the environment against the topology of the graph.

Each directed edge $e \in \mathcal{E}$ indicates connectivity between its endpoints' nodes, and is labelled with the navigational behaviour needed to traverse between these nodes. For each behaviour graph, the navigational behaviours are drawn from a predefined set of N behaviours. While our representation uses the set of behaviours implemented by the DECISION controller [1], in principle the behaviour graph representation can be easily adapted to represent different sets of behaviours.

An additional structural constraint enforced in behaviour graphs is that for any given node v, each outgoing edge from v is required to have a distinct behaviour. Intuitively, this constraint helps to avoid the ambiguity of having different outgoing edges labelled with the same behaviour leading to different destinations.

IV. Building and Navigating with Behaviour Graphs

Figure 1 describes a system that first constructs behaviour graphs from commonly available 2D 'maps' of unseen environments, then employs these behaviour graphs to navigate in the unseen environments. We refer to these as the *offline map-reading* and *online behaviour-based navigation* subsystems respectively.

A. Offline map-reading

The goal of the offline map-reading subsystem is to parse various top-down 2D 'map' representations into behaviour graphs. This process can be decomposed into 2 steps: (1) node prediction, followed by (2) edge prediction using the previously predicted nodes. We design two neural networks ϕ_{node} and ϕ_{edge} to be used for each step respectively. This subsystem is thus a learnable pipeline that can be trained to adapt to different 'map' types - e.g. floor-plans, hand-drawn maps or satellite maps.

Node prediction: The node prediction algorithm takes a 2D 'map' as input, and outputs the pixel locations of the changepoints in the 'map'. It makes use of ϕ_{node}, a CNN that takes in a 2D patch from the 'map' and outputs the likelihood that the patch centre is a changepoint. ϕ_{node} is learned from a set of 'maps' manually annotated with behaviour graphs, and is trained to classify annotated changepoints with the cross-entropy loss. The node prediction algorithm works by first creating a dense regular grid of points across the 'map', extracting and scoring patches centred on each point using ϕ_{node}, then

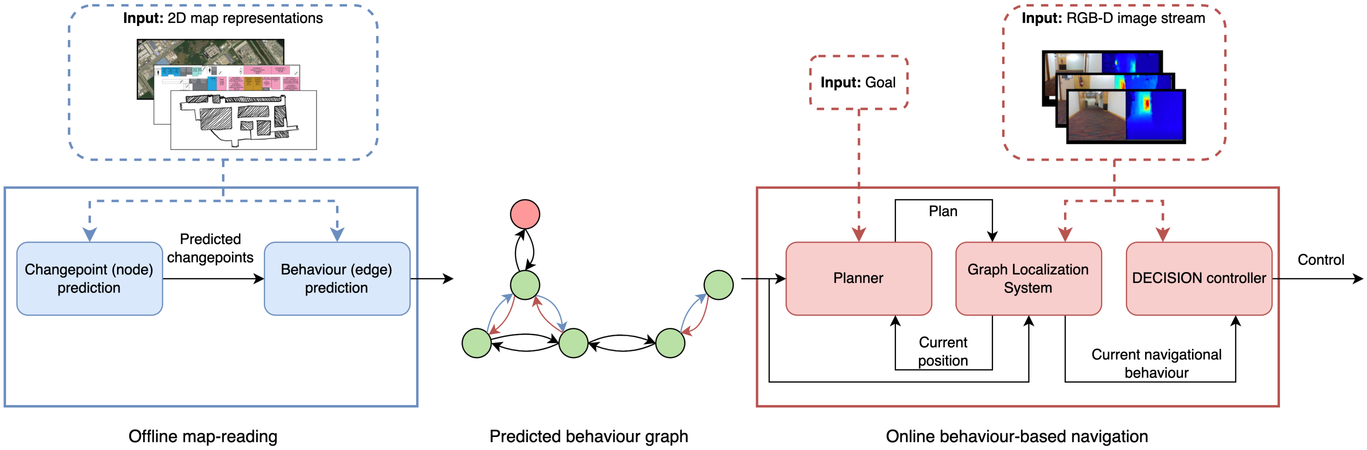

Fig. 1. Overall system architecture. Behaviour graphs are constructed first through *offline map-reading*, then used for *online behaviour-based navigation*.

thresholding the points and performing a clustering-based non-maximal suppression step. The points remaining at the end are the detected changepoints in the 'map'.

Edge prediction: The edge prediction algorithm takes as input a 2D 'map' along with the predicted changepoints on this 'map', and outputs the predicted edges of the behaviour graph. It makes use of ϕ_{edge}, a CNN that takes as input a patch from the 'map' centred on the ith predicted changepoint p_i along with the locations of p_i's K-closest neighbouring nodes, and outputs the set of outgoing edges from p_i. ϕ_{edge} predicts a $K \times N$ cost matrix $\mathcal{C}$, where c_{kn} is a cost representing how likely it is that an edge exists between p_i and neighbour node k with behaviour n. To enforce the behaviour graph's structural constraint, we follow [8] and run the Sinkhorn algorithm with dustbin scores, then threshold the output matrix to obtain the predicted outgoing edges from p_i. To predict all of the edges in the behaviour graph, we run ϕ_{edge} on every predicted changepoint and destination node in the 'map'.

B. Online behaviour-based navigation

Given a behaviour graph of the unseen environment, the online behaviour-based navigation subsystem aims to plan a path to a specified goal node, and navigate the robot safely there. The subsystem comprises 3 main components: the controller, the planner and the graph localisation system.

Controller: The robot's navigational behaviours are implemented in the controller, and we employ the DECISION controller [1] which provides 3 behaviours: {`turn-left`, `go-forward`, `turn-right`}. Each behaviour in the DECISION controller takes in a stream of RGB observations, and generates linear and angular velocity commands.

Planner: The planner takes in a goal node - either specified by node index, or by semantic label - and finds a path from the current location to it with Dijkstra. This path is represented as a sequence of navigational behaviours for the robot to execute, e.g. {`go-straight`, `turn-left`, `go-straight`, ...}.

Graph localisation system: We adapt Graph Localisation Networks (GLN) [3] to localise on the behaviour graph. In GLNs, localisation is defined as finding which *edge* in the graph we are on. This is better defined since the robot may not always be near a node, but will always be executing an action on an edge while navigating. The GLN takes as input an RGB-D image stream, along with a crop of the behaviour graph around the robot's last known position, and outputs the current edge the robot is on. We adapt the GLN to also predict our closeness to the next changepoint, allowing us to fluidly switch between behaviours when navigating. The GLN consists of a CNN that first converts a temporal sequence of RGB images into a feature vector, then passes this feature to a GNN which performs several rounds of message-passing over the behaviour graph crop. Intuitively, the GLN is matching the layout and navigational affordances of the environment observed from the RGB sequence against the behaviour graph's topology to localise the robot.

V. Experiments

We present preliminary results that answer the following questions: **(Q1)** How well can the offline map-reading subsystem extract behaviour graphs from a variety of 'map' types? **(Q2)** Can the extracted behaviour graphs be effectively used for robot navigation?

A. Evaluating offline map-reading

We test the offline map-reading system on hand-drawn maps (HM), floor-plans (FP) and satellite maps (SM), and compare its predictions against manually annotated ground truth behaviour graphs. Precision and recall results for both node and edge predictions are shown in Table I.

Node predictions generally perform well across all 'map' types. We observe that node prediction reliably identifies changepoints at sharp turnings and junctions. A large proportion of the failure cases occur due to under- or over-prediction of changepoints in open areas, where the structure of the environment and the behaviours needed to navigate it are not as well-defined. In the case of edge prediction, many of the failure cases occur because the edge's behaviour label is wrongly assigned, even though the edge itself is

TABLE I
CLASSIFICATION PERFORMANCE OF NODE/EDGE PREDICTION

Metrics	**Changepoints (nodes)**			**Behaviours (edges)**		
	HM	*FP*	*SM*	*HM*	*FP*	*SM*
Precision	0.848	0.732	0.865	0.667	0.630	0.761
Recall	0.975	0.779	0.621	0.535	0.494	0.662

TABLE II
EVALUATING NAVIGATION PERFORMANCE WITH PREDICTED BEHAVIOUR GRAPHS FROM HAND-DRAWN MAPS (HM) AND FLOORPLANS (FP)

Test settings		*SR-HL*	*PC-HL*	*SR-Nav*	*PC-Nav*
Easy	HM	80.0	90.0	80.0	90.0
	FP	68.8	78.1	62.5	71.9
Med	HM	75.0	87.5	62.5	75.0
	FP	37.5	65.6	37.5	65.6
Hard	HM	50.0	85.0	50.0	85.0
	FP	50.0	80.0	50.0	80.0

correctly predicted. In particular, a large proportion of such cases involves a `go-forward` behaviour being confused with a `turn` behaviour. In the next section, we show that despite some behaviour mis-classifications, the behaviour graphs retain enough useful information to enable effective navigation.

B. Evaluating real-world navigation

We deploy online behaviour-based navigation on a Spot quadruped with Intel Realsense cameras and an AGX Xavier. The system undergoes testing in an indoor office setting using graphs inferred by the offline map-reading subsystem. The tests have 3 difficulty levels: *Easy* (2-3 changepoints, 10-25m), *Medium* (4-5 changepoints, 30-50m), *Hard* (6-10 changepoints, 50-100m). We evaluate performance using Success Rate (SR) and Plan Completion (PC) [3]. We report SR and PC metrics for the high-level planning and localisation modules (HL) and the entire navigation stack (Nav). For HL metrics, the task terminates only if an incorrect behaviour is issued or localisation fails, and we manually intervene if the controller fails to execute the commanded behaviour correctly. In Nav metrics, the task also terminates if the controller fails.

We find that PC is high across all map types and difficulty levels, indicating that the system is able to successfully switch behaviours most of the time, and succeeds in navigating most of the way on the test routes. This also suggests that the noisy inferred graphs with mis-classified behaviours can still be effective representations for navigation.

VI. CONCLUSION

We proposed *behaviour graphs*, a compact, graphical, semantic-contextual representation, where nodes are coarse locations and edges represent the navigational behaviours used to transit between the nodes. We proposed a ‘map-reading’ pipeline to extract such graphs from commonly available ‘maps’, and showed its efficacy on a variety of ‘map’ types. Finally we demonstrated that behaviour graphs enable effective navigation on a real robot, even without metric representations or positioning.

REFERENCES

[1] Bo Ai, Wei Gao, Vinay, and David Hsu. Deep visual navigation under partial observability. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9439–9446, 2022. doi: 10.1109/ICRA46639.2022.9811598.

[2] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *IEEE Transactions on Robotics*, 32, 06 2016. doi: 10.1109/TRO.2016.2624754.

[3] Kevin Chen, Juan Pablo de Vicente, Gabriel Sepulveda, Fei Xia, Alvaro Soto, Marynel VÃ¡zquez, and Silvio Savarese. A behavioral approach to visual navigation with graph localization networks. In *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019. doi: 10.15607/RSS.2019.XV.010.

[4] Felipe Codevilla, Matthias Müller, Antonio M. López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 1–9. IEEE, 2018. doi: 10.1109/ICRA.2018.8460487. URL https://doi.org/10.1109/ICRA.2018.8460487.

[5] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto. Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery. *IEEE Robotics and Automation Letters*, 4(3):3037–3044, July 2019. ISSN 2377-3766. doi: 10.1109/LRA.2019.2923960.

[6] Nathan Hughes, Yun Chang, Siyi Hu, Rajat Talak, Rumaisa Abdulhai, Jared Strader, and Luca Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems, 2023.

[7] Benjamin Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 04 2000. doi: 10.1016/S0004-3702(00)00017-5.

[8] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[9] Gabriel Sepulveda, Juan Carlos Niebles, and Álvaro Soto. A deep learning based behavioral approach to indoor autonomous navigation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4646–4653, 2018.

[10] Maks Sorokin, Jie Tan, C. Karen Liu, and Sehoon Ha. Learning to navigate sidewalks in outdoor environments. *IEEE Robotics and Automation Letters*, 7(2):3906–3913, 2022. doi: 10.1109/LRA.2022.3145947.

[11] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning, 2016.