Biological Cybernetics (2019) 113:515–545
https://doi.org/10.1007/s00422-019-00806-9

Biological Cybernetics

ORIGINAL ARTICLE

NeuroSLAM: a brain-inspired SLAM system for 3D environments

Fangwen Yu[1,2] · **Jianga Shang**[1] · **Youjian Hu**[1] · **Michael Milford**[2]

Received: 29 March 2019 / Accepted: 14 September 2019 / Published online: 30 September 2019

Abstract
Roboticists have long drawn inspiration from nature to develop navigation and simultaneous localization and mapping (SLAM) systems such as RatSLAM. Animals such as birds and bats possess superlative navigation capabilities, robustly navigating over large, three-dimensional environments, leveraging an internal neural representation of space combined with external sensory cues and self-motion cues. This paper presents a novel neuro-inspired 4DoF (degrees of freedom) SLAM system named NeuroSLAM, based upon computational models of 3D grid cells and multilayered head direction cells, integrated with a vision system that provides external visual cues and self-motion cues. NeuroSLAM's neural network activity drives the creation of a multilayered graphical experience map in a real time, enabling relocalization and loop closure through sequences of familiar local visual cues. A multilayered experience map relaxation algorithm is used to correct cumulative errors in path integration after loop closure. Using both synthetic and real-world datasets comprising complex, multilayered indoor and outdoor environments, we demonstrate NeuroSLAM consistently producing topologically correct three-dimensional maps.

Keywords Bio-inspired robotics · Brain-inspired navigation · Simultaneous localization and mapping (SLAM) · 3D grid cells · Multilayered head direction cells

1 Introduction

Navigating in three-dimensional environments is a critical capability for many current or prospective robotic tasks, such as rescue, delivery and exploration. Operation in these environments involves a number of challenges including onboard compute, lack of cloud access, power consumption restrictions and cost pressures (Cadena et al. 2016; Bellingham et al. 2018): Current advanced systems typically employ a multi-sensor suite of vision, range, inertial and GNSS (global navigation satellite system) sensors, combined with a probabilistic mapping or simultaneous localization and mapping (SLAM) back-end (Saputra et al. 2018).

In contrast to current robotic technologies, many animals such as bats robustly map and navigate in a range of three-dimensional environments (Jeffery et al. 2013; Finkelstein et al. 2016). The location in 3D space is estimated by combining local visual cues and self-motion cues for spatial navigation in the mammalian brain (Milford and Schulz 2014; Finkelstein et al. 2016; Campbell et al. 2018). Some neural basis of navigation has been discovered in the mammalian brain, including place cells (O'Keefe and Dostrovsky 1971), grid cells (Hafting et al. 2005), head direction cells (Taube et al. 1990), boundary/border cells (Solstad et al. 2008; Lever et al. 2009), speed cells (Kropff et al. 2015), etc. (Moser et al. 2017). Together, an internal cognitive map is generated according to local visual cues and self-motion cues. (Jeffery et al. 2013, 2016; Evans et al. 2016; Cope et al. 2017; Campbell et al. 2018; Bjerknes et al. 2018).

Communicated by Benjamin Lindner.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00422-019-00806-9) contains supplementary material, which is available to authorized users.

✉ Jianga Shang
jgshang@cug.edu.cn

Fangwen Yu
yufangwen@cug.edu.cn

Youjian Hu
huyj@cug.edu.cn

Michael Milford
michael.milford@qut.edu.au

[1] Faculty of Information Engineering, China University of Geosciences and National Engineering Research Center for Geographic Information System, Wuhan 430074, China

[2] Science and Engineering Faculty, Queensland University of Technology and Australian Centre for Robotic Vision, Brisbane, QLD 4000, Australia

With the initial discovery and understanding of the 2D spatial neural mechanism in the brain encoded by place and head direction cells, some neural navigational models have been developed and applied to robot navigation. For example, a navigational model based on head direction cells and place cells was developed, which was deployed on the Khepera robot operating in a small 2D area (Arleo and Gerstner 2000). Furthermore, in order to support large-scale persistent navigation and mapping, Milford et al. (2004), Milford and Wyeth (2008) and Milford and Wyeth (2010) developed a computational model called RatSLAM, a rodent brain-inspired SLAM algorithm. RatSLAM has successfully mapped an entire suburb in a 2D map and navigated in an office environment over two weeks. Most recently, several novel approaches have been developed based on several types of neural network models and neuromorphic hardware (Banino et al. 2018; Tang and Michmizos 2018; Zhou et al. 2018; Kreiser et al. 2018a, b).

There has to date been relatively little work on developing biologically inspired mapping models capable of functioning in 3D, rather than 2D environments. Part of this is due to the relatively recent inroads into understanding 3D spatial representations in the brain; neuroscientists have recently found evidence for the neural basis of 3D navigational neural representation in freely flying bats, rats and humans, including 3D place cells (Yartsev and Ulanovsky 2013; Kim et al. 2017; Wohlgemuth et al. 2018), 3D head direction cells (Finkelstein et al. 2015; Laurens et al. 2016; Page et al. 2018; Shinder and Taube 2019) and 3D grid cells (Finkelstein et al. 2016; Kim and Maguire 2019; Casali et al. 2019). Hayman et al. (2015) and Jeffery et al. (2015) proposed several mathematical models of these 3D spatial neural cells and analyzed their properties and limitations in representing 3D space. Page et al. (2018) proposed a 3D rotation rule with dual axis for representing 3D head direction. Casali et al. (2019) found the novel spatial encoding properties of grid cell firing fields in vertical space. Some research has also closed the loop back to neuroscience in an attempt to aid neuroscientists in interpreting neurobiological recordings (Llofriu et al. 2015; Gianelli et al. 2018; Gaussier et al. 2019).

In this paper, we present a novel biologically inspired mapping system with 4 degrees of freedom, enabling it to map and localize in 3D, rather than 2D, environments. The core system draws some of its inspiration from previous 2D-only brain-based mapping systems including RatSLAM (Milford et al. 2004; Milford and Wyeth 2008, 2010), but makes a range of new contributions as follows:

- Firstly, we propose a novel neuro-inspired model for mapping and localization in a large, real-world three-dimensional environments, which is to the best of our knowledge, the first work to do so.
- Secondly, we develop a functional computational model of *conjunctive pose cells* consisting of 3D grid cells and multilayered head direction cells for representing a 4DoF pose (x, y, z, yaw).
- Thirdly, we propose a novel multilayered graphical experience map combining the local view cells, 3D grid cells, multilayered head direction cells and 3D visual odometry.
- Finally, we present three new 3D mapping real-world and synthetic datasets comprising both outdoor and indoor environments and evaluate NeuroSLAM's performance on them.

The paper is organized as follows: Sect. 2 discusses the conventional visual SLAM and brain-inspired SLAM. Section 3 describes the current understanding of 3D spatial neural representation in the brain and provides a background for the problem. Section 4 presents the architecture and detailed models of our system—NeuroSLAM—to build the neural mechanism of 3D spatial representation in robots. Section 5 describes the methodology of experiments for investigating the performance of NeuroSLAM. Section 6 presents experimental results demonstrating the mapping and localization performance of the system in 3D space. Section 7 discusses and concludes the results of the study, revealing the insights gained regarding the benefits and drawbacks of developing and deploying a neuro-inspired 3D SLAM system.

2 State of the art in the SLAM for 3D environments

In this section, we briefly review the conventional visual SLAM and brain-inspired SLAM for 3D environments. A 3D SLAM system enables a robot to explore in an unknown 3D environment from an arbitrary initial 3D location. Meanwhile, it can build a 3D spatial map incrementally. The 3D spatial map is also used to compute the robot's 3D pose simultaneously (Dissanayake et al. 2001; Thrun and Leonard 2008). Approaches to solve the problem of 3D SLAM broadly fall into two classes. The primary set of approaches is typically geometric in nature and is driven by optimization or probabilistic filters, for instance, graph optimization, particle filters or extended Kalman filters (EKF) (Cadena et al. 2016). A second set of approaches to SLAM has been based on inspiration from biological mapping and localization systems. These biologically inspired methods also fall into two classes. One set of approaches is based on the navigational behavior strategies of ants, bees and insects (Sabo et al. 2016, 2017; Stone et al. 2016; Dupeyroux et al. 2019). Another set of approaches is based on neural navigational mechanisms. In this paper, we mainly focus on the approaches based on 3D neural navigational mechanisms.

In the following sections, we review both conventional 3D visual SLAM and prior brain-inspired SLAM systems.

2.1 Conventional 3D visual SLAM

The robot must build up a 3D spatial map to navigate effectively in 3D environments. Generally, four classes of spatial representation, including geometrical, topological, semantic and hybrid maps, are used in modeling spaces. In recent years, a popular and essential topic has been 3D visual SLAM due to the decreasing cost of cameras and the similarity to mammalian 3D visual perception (Milford 2013; Welchman 2016; Naseer et al. 2018). Monocular, stereo, omnidirectional, RGB-D cameras and 3D laser range finders are among the well-known sensors used for 3D SLAM (Faessler et al. 2016; Saputra et al. 2018). Some advanced approaches to 3D visual SLAM and 3D visual odometry are given in Tables 1 and 2. Noticeable approaches include ORB-SLAM (Mur-Artal and Tardós 2017; Mur-Artal et al. 2015), PTAM (Klein and Murray 2007), LSD-SLAM (Engel et al. 2014), SVO (Forster et al. 2014, 2017) and DSO (Engel et al. 2018). Recently, some novel approaches have been proposed based on biologically analogous event cameras, such as EVO (Rebecq et al. 2017) and event camera-based SLAM (Vidal et al. 2018).

In addition, some approaches have focused on the loop closure component of the SLAM problem, such as FrameSLAM (Konolige and Agrawal 2008), FAB-MAP (Cummins and Newman 2008; Paul and Newman 2010), SeqSLAM (Milford and Wyeth 2012) and CAT-SLAM (Maddern et al. 2012). More details about place recognition and loop closure can be found in the survey paper by Lowry et al. (2016).

Many state-of-the-art SLAM solutions for building spatial maps work well in static, structured and 3D environments (Cadena et al. 2016). In order to estimate 3D pose of the robot in large 3D environments, many optimization and filter algorithms have been proposed. However, many of these algorithms require significant computational resources, costly sensors and the assumption of static environments. Furthermore, bad data association often impairs their application to complex 3D environments (Cadena et al. 2016; Bellingham et al. 2018). Overall, the SLAM in unstructured, large-scale and 3D open environments is still an open challenging problem. We investigate the feasibility of a bio-inspired, hybrid spatial representation approach combining topological and metric information for 3D environments in this study.

2.2 Brain-inspired SLAM

Mammalian animals can find food, return to their nest and find social mates by using their navigation capabilities. With the discovery of and improvements in our understanding of neural mechanisms in the brain, some navigational neural network models have been developed and applied into the robot navigation in 2D areas. For instance, a navigational computational model of head direction cells and place cells was developed, which was deployed on Khepera robot operating in a small 2D area (Arleo and Gerstner 2000). In addition, a robot architecture with the capability of spatial navigation was developed by Barrera and Weitzenfeld (2008). In order to support large-scale persistent navigation and mapping, a bio-inspired SLAM model, called RatSLAM, was developed (Milford et al. 2004; Milford and Wyeth 2008, 2010). The model loosely imitates parts of the rodent brain. RatSLAM has successfully mapped an entire suburb in a 2D map and navigated in a 2D office environment over two weeks.

Most recent expansional approaches based on the RatSLAM model have been developed, such as BatSLAM (Steckel and Peremans 2013) using the sonar sensing modality. Tang et al. (2018) integrated an episodic memory module for processing the context in navigational tasks. Furthermore, Milford et al. (2011a) and Milford et al. (2011b) improved the vision system to solve SLAM problem in 2.5D environments without changing the core model of RatSLAM. Silveira et al. (2013) and Silveira et al. (2015) explored the SLAM problem in a 3D underwater environment by expanding the RatSLAM model using a 3D place cell model, but they do not represent metric and directional information.

In addition, some novel models and approaches have been developed based on place cells (PC), head direction cells (HDC) and grid cells (GC) with various types of neural networks, such as continuous attractor neural networks (CANN), deep neural networks (DNN) and spiking neural networks (SNN), as given in Table 3. Several approaches have used novel sensors, such as event-based camera and RGB-D sensors, as well as neuromorphic hardware, such as Kreiser et al. (2018a, b).

Many approaches inspired by the spatial representation in the brain have been developed for 2D SLAM in robots. However, few if any have tackled the challenging problem of 3D SLAM in challenging real-world environments based on the 3D spatial neural representation in the mammalian brain. Until relatively recently, this focus on 2D has surely been in part due to relatively little being known about the neural substrates underlying 3D navigation. However, recent discoveries of 3D navigational neural representation in flying bats and the human brain have provided some new sources of inspiration for modelers and roboticists. In this paper, we focus on developing a neural model for 3D spatial representation in order to provide a bio-inspired SLAM capability in 3D environments.

Table 1 Comparison of 3D SLAM methods. The type of motion is noted with labels '2D' (3DoF motion) and '3D' (free 6DoF motion in 3D space)

	2D/3D	Sensors	Algorithm	Loop closure	Direct/indirect	Dense/sparse	Large scale	References
ORB-SLAM2	3D	Monocular, Stereo, RGB-D	Graph optimization	Yes	Indirect	Sparse	Yes	Mur-Artal et al. (2015), Mur-Artal and Tardós (2017)
LSD-SLAM	3D	Monocular	Graph optimization	Yes	Direct	Semi-dense	Yes	Engel et al. (2014)
MonoSLAM	3D	Monocular	Extended Kalman Filters	Yes	Indirect	Sparse	Yes	Davison et al. (2007)
PTAM	3D	Monocular	Graph optimization	Yes	Indirect	Sparse	Yes	Klein and Murray (2007)
OKVIS	3D	Stereo, IMU	Information filter	Yes	Indirect	Sparse	Yes	Lynen et al. (2016)
Event SLAM	3D	Event camera, IMU, Monocular	Bayesian filter	Yes	Indirect	Sparse	Yes	Vidal et al. (2018), Gallego et al. (2018)
GraphSLAM	3D	Monocular, IMU	Graph optimization	Yes	Indirect	Sparse	Yes	Thrun and Montemerlo (2006)
FastSLAM	3D	Laser	Particle filters	Yes	Indirect	Dense	Yes	Montemerlo et al. (2002)
AcousticSLAM	3D	Acoustic	Probability hypothesis Density filter	No	–	Dense	No	Evers and Naylor (2018)
CVI-SLAM	3D	Monocular, IMU	Kalman filters	Yes	Indirect	Sparse	Yes	Karrer et al. (2018)
Stereo SLAM	3D	Stereo camera	Graph optimization	Yes	Direct	Semi-dense	Yes	Krombach et al. (2018)
Laser-SLAM	3D	Laser	Graph optimization	Yes	Indirect	Dense	Yes	Droeschel et al. (2017), Behley and Stachniss (2018)
RGB-D-SLAM	3D	RGB-D	Graph optimization	Yes	Indirect	Dense	Yes	Henry et al. (2012), Endres et al. (2014)
DTAM	3D	RGB-D	Primal–dual	Yes	Direct	Dense	Yes	Newcombe et al. (2011)

Table 2 Comparison of 3D visual odometry approaches

	2D/3D	Sensors	Algorithm	Direct/indirect	Dense/sparse	Large scale	References
SVO	3D	Perspective, fisheye, catadioptric camera	Graph optimization	Semi-direct	Dense	Yes	Forster et al. (2017)
DSO	3D	Perspective camera	Information filter	Direct	Sparse	Yes	Engel et al. (2018)
OmniDSO	3D	Fisheye camera	Information filter	Direct	Sparse	Yes	Matsuki et al. 2018
EVO	3D	Event camera	Kalman filters	Indirect	Sparse	Yes	Rebecq et al. (2017)
LibViso2	3D	Perspective camera	Kalman filters	Indirect	Dense	Yes	Geiger et al. (2011)
VINS-Mono	3D	Monocular, IMU	Graph optimization	Indirect	Dense	Yes	Qin et al. (2018)

3 3D spatial representation in the mammalian brain

In this section, we describe the current understanding of 3D spatial neural representation in the brain and provide some background context for the NeuroSLAM model. After brief review of some key navigational neural cells in the brain, we mainly describe the properties of the 3D grid cells and the head direction cells. We then describe the multidimensional attractor neural network we have developed to model the 3D grid cells and the multilayered head direction cells.

Neuroscientists have discovered some neural basis of neural spatial representation in the mammalian brain which can support 2D navigation (Moser et al. 2017). However, many animals are able to navigate in 3D space, but until recently, we still knew very little about 3D spatial representation in the mammalian brain. In recent years, neuroscientists have found some neural basis of 3D navigational neural representation in freely flying bats and rats, including 3D place cells (Yartsev and Ulanovsky 2013; Wohlgemuth et al. 2018), 3D head direction cells (Finkelstein et al. 2015; Laurens et al. 2016; Page et al. 2018; Shinder and Taube 2019) and 3D grid cells (Finkelstein et al. 2016; Casali et al. 2019). In addition, the latest investigations have shown that 3D place cells, 3D head direction cells and 3D grid cells exist in the human brain (Kim et al. 2017; Kim and Maguire 2018a, b, 2019). The 3D place cells discharge selectively when mammals pass through a certain 3D spatial location, which form a metric map in all three dimensions (Yartsev and Ulanovsky 2013; Finkelstein et al. 2016). The 3D head direction cells respond to a particular combination of azimuth x pitch thus representing the direction of the head vector in 3D space (Finkelstein et al. 2015, 2016). The 3D grid cells would exhibit regular 3D lattice pattern, which represent 3D position, direction and metric information for 3D path integration (Finkelstein et al. 2016; Jeffery et al. 2015). Jeffery et al. (2015) and Hayman et al. (2015) presented several mathematical models of these 3D spatial neural cells and analyzed the properties and constraints in representing 3D space. Page et al. (2018) proposed a 3D rotation rule with dual axis for representing 3D head direction. Casali et al. (2019) found the spatial encoding properties of the grid cells in vertical space. Soman et al. (2018) modeled the 3D spatial neural cells based on a hierarchical network. In this paper, we represent the 4DoF pose by combining models of 3D grid cells and head direction cells. The properties of these cells are described in the following section.

3.1 3D grid cells

Grid cells are a type of neurons in the mammalian brain which have a periodic hexagonal pattern of firing fields. This property is independent of the direction and speed of a moving

Table 3 Comparison of brain-inspired SLAM and navigation systems. The place cells, grid cells and head direction cells are abbreviated as PC, GC and HDC, respectively. The continuous attractor neural network, deep neural network and spiking neural network are abbreviated as CANN, DNN and SNN, respectively

	Cells	Dimension	Sensor	Network	Neuromorphic hardware	Experimental range	References
Arleo and Gerstner (2000)	PC HDC	2D	Monocular camera	Multilayer network	No	Small	Arleo and Gerstner (2000)
RatSLAM	Pose cells	2D	Monocular camera	CANN	No	Large	Milford et al. (2004), Milford and Wyeth (2008, 2010)
Meyer et al. (2005)	PC	2D	Monocular camera	CANN	No	Small	Meyer et al. (2005)
Giovannangeli and Gaussier (2008)	PC	2D	Omnidirectional camera	NN	No	Small	Giovannangeli and Gaussier (2008)
BatSLAM	Pose cells	2D	Sonar	CANN	No	Small	Steckel and Peremans (2013)
Jauffret et al. (2015)	GC, PC	2D	Fisheye camera	NN	No	Small	Jauffret et al. (2015)
Tang et al. (2018)	Pose cells, Episodic memory	2D	Stereo camera, magnetic compass, IR sensors	CANN, SNN	No	Small	Tang et al. (2018)
Zeng and Si (2017)	HDC, Conjunctive GC	2D	Monocular camera	CANN	No	Small	Zeng and Si (2017)
Banino et al. (2018)	GC PC HDC	2D	Monocular camera	DNN	No	Small	Banino et al. (2018)
Gridbot	PC, HDC, GC, Border cell	2D	RGB-D	SNN	No	Small	Tang and Michmizos (2018)
Kreiser et al. (2018a, b)	HDC	2D	Event-based camera (DVS), IMU	SNN	Yes	Small	Kreiser et al. (2018a, b)
Zhou et al. (2018)	HDC PC	2D	Monocular camera	Hierarchical SFA network	No	Small	Zhou et al. (2018)
Milford et al. (2011b)	Pose cells	2.5D	Fisheye camera	CANN	No	Large	Milford et al. (2011b)
DolphinSLAM	3D Place cells	3D	Sonar camera	CANN	No	Small	Silveira et al. (2015)

animal (Hafting et al. 2005). Neuroscientists thereby thought that grid cells can provide a metric spatial representation for navigation. Furthermore, some investigation revealed that the grid cell network may perform a path integration based on self-motion cumulatively (Hafting et al. 2005; Burak and Fiete 2009). Recently, Finkelstein et al. (2016) predicted that 3D grid cells existing in the bat brain. Kim and Maguire (2019) provided some key evidence for the existence of 3D grid cells in the human brain. The models of the hexagonal close packing (HCP) and the face-centered cubic lattice (FCC) are proposed to organize 3D grid cells (Jeffery et al. 2013, 2015; Horiuchi and Moss 2015; Laurens and Angelaki 2018; Kim and Maguire 2019). In this paper, we use the 3D grid cell model to represent 3D position and metric information for 3D path integration.

3.2 Head direction cells

Head direction cells are a type of neurons in the mammalian brain. They can discharge when the animal is oriented in a particular direction (Taube et al. 1990). Additionally, the latest study revealed that 3D head direction cells can represent the direction of the animal with yaw, pitch, roll or their combination in 3D space (Finkelstein et al. 2015, 2018; Kim and Maguire 2018b). Some experiments (Stackman et al. 2000) have shown that the head direction cells can represent global direction information during 3D navigation in distinct floors. In this paper, we only take azimuth into consideration for representing a 4DoF pose, and a multilayered head direction cell model is used to represent the robotic orientation.

3.3 Multidimensional continuous attractor network

Multidimensional continuous attractor network (MD-CAN) is a significant approach to modeling spatial neural cells (Samsonovich and McNaughton 1997; Burak and Fiete 2009; Mulas et al. 2016; Jeffery et al. 2016; Laurens and Angelaki 2018). The MD-CAN is a type of neural network with weighted excitatory and inhibitory connections (Shipston-Sharman et al. 2016). The MD-CAN has many recurrent connections which cause the network to converge over time to certain stable states (attractors, activity packets or bumps) in the absence of external input (Milford and Wyeth 2008). The MD-CAN operates by updating the neural activity. Unlike most neural networks, it does not change the value of the weighted connections (Milford and Wyeth 2010). Each neural unit in the MD-CAN has a continuous activation value between zero and one. When the robot approaches a spatial location, the activation value of the associated neural unit increases. Their properties are significantly different from the usual probabilistic representations found in conventional SLAM algorithms. In this study, the 2D MD-CAN and 3D MD-CAN are used to represent the multilayered head direction cell model and 3D grid cell model, respectively.

4 Our approach: NeuroSLAM

In this section, we describe the detailed architecture and the components of the NeuroSLAM system: the conjunctive pose cell model, the multilayered experience map and the vision module. After giving an overview of the system, we present the detailed computational models of the 3D grid cells and the multilayered head direction cells and their network operation process with attractor dynamics, path integration and local view calibration. We then describe the multilayered experience map creation and relaxation processes. The section concludes with a description of the vision system for providing external visual cues and self-motion cues. The NeuroSLAM code is available at https://github.com/cognav/NeuroSLAM.git.

4.1 Architecture

The architecture of the NeuroSLAM system is shown in Fig. 1. The robot's state of 4DoF pose (x, y, z, yaw) in 3D environments is represented by the activity in the 3D grid cell network and the multilayered head direction cell network, conjunctively. The conjunctive pose cell network performs path integration on the basis of the self-motion cues and performs calibration based on the local visual cues. The approaches to creation and relaxation of multilayered graphical experience map are based on the combination of local view cells, conjunctive pose cells and 3D visual odometry. The algorithm of the overall main program is described in Algorithm 1 (Online Appendix).

We update the activity of the MD-CAN for the conjunctive pose cells according to self-motion cues. The activity is calibrated by local view cues when seeing familiar scenes. Wrap-around connections connect each boundary of the MD-CAN to its opposing boundary. The path integration is performed in the MD-CAN by injecting activity into the conjunctive pose cells. Therefore, the current activity packets are shifted according to the 3D visual odometry. External visual cue activates local view cells if the cue is familiar. The activity is injected into the particular conjunctive pose cells by the activated local view cell through associative learning. The local view and self-motion cues are generated from successive images collected with a perspective or panoramic camera.

The multilayered experience map is used to represent 3D spatial experience. A particular 3D spatial experience is encoded by a conjunctive code combining a set of information from the local view cells, the multilayered head

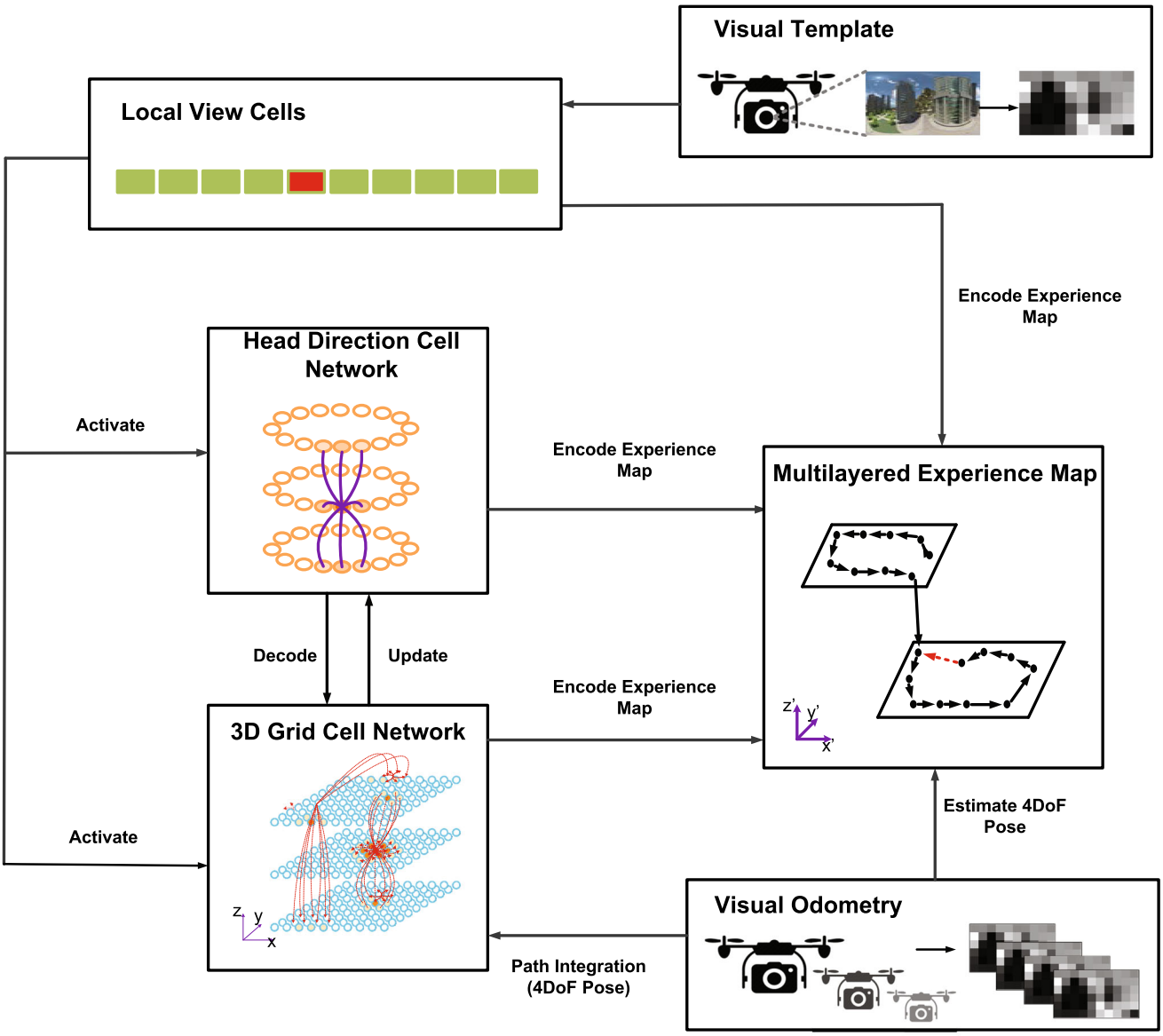

Fig. 1 The NeuroSLAM architecture. The system consists of conjunctive pose cells combining the 3D grid cells and multilayered head direction cells, the multilayered experience map and vision modules. The conjunctive pose cell network performs path integration based on the local view cues and self-motion information. The distinct scenes are encoded by the local view cells. The output from the local view cells, the 3D grid cells, the multilayered head direction cells and the 3D visual odometry drives the creation of a 3D multilayered experience map, which is a hybrid spatial representation with a topological and locally metric 3D graphical map of the 3D environment

direction cells, the 3D grid cells and the self-motion cues. We define an individual 3D spatial experience by the pattern of activity in the local view cells and the conjunctive pose cells. When the pattern changes, we create a new 3D spatial experience. Meanwhile, a new transition from the old experience to the new experience is also created. The transition contains the change of 4DoF pose between the two experiences. When the robot moves in a new environment, new 3D spatial experiences and their transitions will form continuously. When the robot revisits a familiar environment, a loop closure occurs if the conjunctive code is the same as in a previous stored codes. The multilayered map relaxation is also performed to keep topological consistent.

The following sections describe the computation process in the 3D grid cell network, the multilayered head direction cell network and the multilayered experience map. We also describe the estimation process for the vision system.

4.2 Conjunctive pose cell model

We use the conjunctive pose cells consisting of 3D grid cells and multilayered head direction cells to represent a 4DoF pose (x, y, z, yaw). In order to improve computing efficiency and reduce the complexity of the system, we simplify the neural model. We do not build a 3D place cell model. Instead, the functional characteristics of place cells are encoded in the 3D grid cell network and the 3D experience map. The 3D location (x, y, z) is represented by 3D grid cells using a 3D MD-CAN. The orientation (yaw) is represented by multilayered head direction cells using a 2D MD-CAN. In this study, we do not take pitch and roll into consideration. We describe more details of the 3D grid cell model and the multilayered head direction cell model in the following sections.

4.2.1 3D grid cell model

The 3D grid cell network is a 3D MD-CAN. It mimics the 3D spatial neural representation in the mammalian brain, as shown in Fig. 2. The model is similar to the FCC (face-centered cubic) model in Jeffery et al. (2015) and Kim and Maguire (2019). The network exhibits regular 3D lattice pattern which represents 3D position, direction and metric information for 3D path integration. It can maintain a robot's 4DoF pose in the absence of external sensory cues inputs. Each area of the 3D environment is encoded by particular neuron in the network. The model can perform 3D path integration based on the self-motion cues. The local view cells are associated with specific 3D grid cells. When the robot sees familiar scenes, the 3D MD-CAN can perform 3D location calibration. The algorithm of 3D grid cell network is described in the Algorithm 2 (Online Appendix).

The 3D grid cells represent the absolute location (x, y, z) in 3D space. The activity matrix $\mathbf{P}^{\mathrm{gc}}$ describes the activity in the 3D grid cells. We update the activity based on three key processes. Firstly, the activity is updated by the attractor dynamics with excitation and inhibition. Then, the activity packets are shifted by 3D path integration based on the translational and rotational velocity provided by 3D visual odometry. Finally, when the robot sees familiar scenes, the activity is updated by local view calibration.

Attractor dynamics

The process of the internal attractor dynamics in the 3D MD-CAN includes three stages. Firstly, parts of 3D grid cells are excited by the local excitatory process. Then, all 3D grid cells are inhibited by the global inhibition process. Finally, the activity of the 3D grid cells is normalized.

Local excitation

We create the excitatory weight matrix $\boldsymbol{\varepsilon}^{\mathrm{gc}}_{u,v,w}$ using a 3D Gaussian distribution. The distance indexes between units are represented by u, v and w. The weight is calculated by

$$\varepsilon^{\mathrm{gc}}_{u,v,w} = \frac{1}{(\delta_x\sqrt{2\pi})}e^{-u^2/(2\delta_x^2)} \cdot \frac{1}{(\delta_y\sqrt{2\pi})}e^{-v^2/(2\delta_y^2)} \cdot \frac{1}{(\delta_z\sqrt{2\pi})}e^{-w^2/(2\delta_z^2)}, \tag{1}$$

where δ_x, δ_y and δ_z are the constants of variance for 3D spatial distributions. The activity change in a 3D grid cell is calculated by

$$\triangle\mathbf{P}^{\mathrm{gc}}_{x,y,z} = \sum_{i}^{n_x}\sum_{j}^{n_y}\sum_{k}^{n_z}\mathbf{P}^{\mathrm{gc}}_{i,j,k}, \tag{2}$$

where the three dimensions of the matrix are n_x, n_y, n_z. The distance indexes are calculated by

$$\begin{aligned} u &= (x-i)(\mathrm{mod}n_x),\\ v &= (y-j)(\mathrm{mod}n_y),\\ w &= (z-k)(\mathrm{mod}n_z). \end{aligned} \tag{3}$$

Global inhibition

Each 3D grid cell inhibits nearby cells by a local inhibitory process. We create an inhibitory weight matrix $\boldsymbol{\psi}^{\mathrm{gc}}_{u,v,w}$ to update the activity during the local inhibitory process. Then, the activity of all 3D grid cells is updated by the global inhibition φ equally. The processes of the local inhibition and the global inhibition are calculated by

$$\triangle\mathbf{P}^{\mathrm{gc}}_{x,y,z} = \sum_{i}^{n_x}\sum_{j}^{n_y}\sum_{k}^{n_z}\mathbf{P}^{\mathrm{gc}}_{i,j,k}\boldsymbol{\psi}^{\mathrm{gc}}_{u,v,w} - \varphi, \tag{4}$$

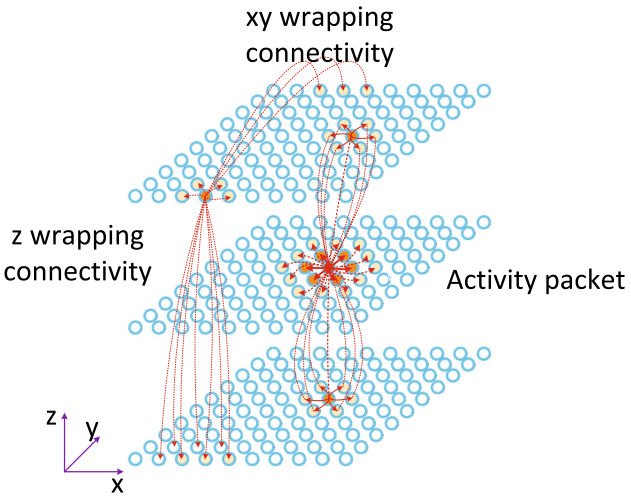

Fig. 2 3D grid cell network model. A 3D MD-CAN (3D cube) represents the 3D grid cells with a stable activity packet. The warp connections of neural units can represent 3D location periodically. Each 3D grid cell is activated when the robot moves in a large-scale space

where we control all values in $\mathbf{P}^{gc}$ to nonnegative values.

Activity normalization

The total activity in 3D grid cells is kept one by activity normalization. The activity, $\mathbf{P}^{gc'}_{x,y,z}$, is calculated by

$$\mathbf{P}^{gc'}_{x,y,z} = \frac{\mathbf{P}^{gc}_{x,y,z}}{\sum_i^{n_x}\sum_j^{n_y}\sum_k^{n_z}\mathbf{P}^{gc}_{i,j,k}}. \tag{5}$$

In the following sections, we describe the update process of the activity in 3D grid cells by 3D path integration and local view calibration.

Path integration

The path integration projects the 3D grid cells activity into nearby cells. The activity is shifted in x, y plane and z dimension according to the translational velocity v and height change velocity v_h along x-, y-, z-axis, respectively, under current head direction in yaw (θ). The activity change ΔU^{gc}_{lmn} is calculated by

$$\Delta U^{gc}_{lmn} = \sum_{x=\delta_{x_0}}^{\delta_{x_0}+1}\sum_{y=\delta_{y_0}}^{\delta_{y_0}+1}\sum_{z=\delta_{z_0}}^{\delta_{z_0}+1}\gamma U^{gc}_{(l+x)(m+y)(n+z)}. \tag{6}$$

The amount of activity injected is determined by two inputs. One is from the product of the sending unit, U^{gc}. Another is from the residue, γ. The residue is calculated according to the fractional parts of the offsets, $\delta_{x_f}, \delta_{y_f}, \delta_{z_f}$.

$$\begin{bmatrix}\delta_{x_0}\\ \delta_{y_0}\\ \delta_{z_0}\end{bmatrix} = \begin{bmatrix}\lfloor k_x v\cos\theta\rfloor\\ \lfloor k_y v\sin\theta\rfloor\\ \lfloor k_z v_h\rfloor\end{bmatrix}, \tag{7}$$

$$\begin{bmatrix}\delta_{x_f}\\ \delta_{y_f}\\ \delta_{z_f}\end{bmatrix} = \begin{bmatrix}\lfloor k_x v\cos\theta - \delta x_0\rfloor\\ \lfloor k_y v\sin\theta - \delta y_0\rfloor\\ \lfloor k_z v_h - \delta z_0\rfloor\end{bmatrix}, \tag{8}$$

where k_x, k_y and k_z are constants for 3D path integration. The γ is calculated by

$$\begin{aligned}\gamma &= f(\delta_{x_f}, x-\delta_{x_0})f(\delta_{y_f}, y-\delta_{y_0})f(\delta_{z_f}, z-\delta_{z_0}),\\ f(a,b) &= \begin{cases}a, & \text{if } b=1,\\ 1-a, & \text{if } b=0.\end{cases}\end{aligned} \tag{9}$$

Local view calibration

The local view calibration resets the accumulative errors in path integration based on the translational and rotational velocity provided by 3D visual odometry. The local view cells are associated with the 3D grid cells and the multilayered head direction cells. When the robot sees familiar view, the prior associations are recalled. We use a vector $\mathbf{V}$ to represent the activity of the local view cells. If the current view is similar to the previous view, the associated local view cell is active.

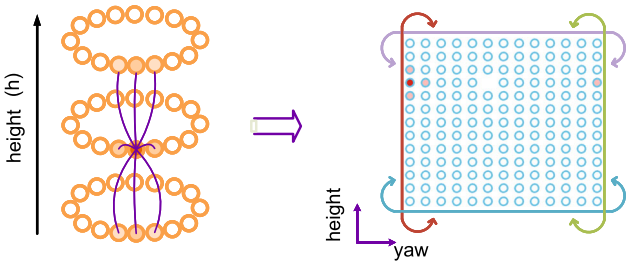

Fig. 3 Multilayered head direction cell network model. The multilayered head direction cells are represented in a MD-CAN (2D matrix). The warp connections of neural units can represent head direction periodically, as shown in the right figure. If the robot's head direction rotates more than one full revolution or moves through a large vertical space, each multilayered head direction cell unit is activated periodically

The connection matrix Ψ stores the learned connections among the local view cell vector, the 3D grid cell matrix and the multilayered head direction cell matrix. We use a modified version of Hebb's law for learning connections. The connection between the $P_{x,y,z,\theta}$ and V_i is calculated by

$$\Psi^{t+1}_{i,x,y,z,\theta} = \max(\tau V_i P_{x,y,z,\theta}, \Psi^{t}_{i,x,y,z,\theta}). \tag{10}$$

The τ is a learning rate. The activity change in 3D grid cells and multilayered head direction cells is calculated by

$$\triangle P_{x,y,z,\theta} = \frac{\delta}{n_{\text{act}}}\sum_i V_i \Psi_{i,x,y,z,\theta}, \tag{11}$$

where the constant δ controls the strength of local view calibration. The n_{act} is the number of active local view cells.

4.2.2 Multilayered head direction cell model

The multilayered head direction cell network is a MD-CAN. It mimics the neural mechanism related to 3D direction cognition of mammals, as shown in Fig. 3. The model is similar to the head direction cell model in Samsonovich and McNaughton (1997) and McNaughton et al. (2006). The MD-CAN can maintain head direction of the robot in 3D space. Each neuron is responsible for mapping a particular direction (yaw) in a specific vertical area. Finkelstein et al. (2015) found there are some cells representing azimuth, pitch, azimuth x pitch and proposed a toroidal model for modeling 3D head direction cells. However, in order to represent 4DoF pose efficiently, we represent azimuth in a 3D vertical space with a multilayered head direction model. The head direction cells are associated with local view cells for direction calibration. The algorithm of multilayered head direction network is described in Algorithm 3 (Online Appendix).

The multilayered head direction cells are arranged in a 2D neural network representing absolute orientation yaw θ corresponding to each vertical area. The cell activity matrix $\mathbf{P}^{hdc}$

describes the activity in the multilayered head direction cells. The activity is updated by three processes. Firstly, the activity in the MD-CAN is updated by the attractor dynamics. Then, the MD-CAN performs a head direction update based upon the output of direction change and height change, obtained from the 3D grid cell network after performing path integration according to the rotational velocity, height change velocity and translational velocity. Finally, when the robot sees a familiar view, the activity in the MD-CAN is updated by the local view calibration process (as described in Sect. 4.2.1).

Attractor dynamics

The process of the internal attractor dynamics in the MD-CAN includes three stages. Firstly, parts of the multilayered head direction cells are excited by the local excitatory process. Then, all multilayered head direction cells are inhibited by the global inhibition process. Finally, the activity in the MD-CAN is normalized.

Local excitation

We create the excitatory weight matrix $\varepsilon^{\text{hdc}}_{u,v}$ using a 2D Gaussian distribution. The distance indexes between units in (θ, h) matrix are represented by u and v. The weight is calculated by

$$\varepsilon^{\text{hdc}}_{u,v} = \frac{1}{(\delta_\theta\sqrt{2\pi})} e^{-u^2/(2\delta_\theta^2)} \cdot \frac{1}{(\delta_h\sqrt{2\pi})} e^{-v^2/(2\delta_h^2)}, \tag{12}$$

where δ_θ and δ_h are the variance constants of the distributions in the orientation and height (θ, h). The activity change in a multilayered head direction cell is calculated by

$$\triangle \mathbf{P}^{\text{hdc}}_{\theta,h} = \sum_{i}^{n_\theta}\sum_{j}^{n_h} \mathbf{P}^{\text{hdc}}_{i,j}, \forall i, j \in u, v, \tag{13}$$

where the matrix dimensions of the multilayered head direction cells are n_θ, n_h. The distance indexes are calculated by

$$\begin{aligned} u &= (\theta - i)(\text{mod} n_\theta), \\ v &= (h - j)(\text{mod} n_h). \end{aligned} \tag{14}$$

Global inhibition

Each multilayered head direction cell inhibits nearby cells by a local inhibition process. Then, all multilayered head direction cells are inhibited by a constant of global inhibition φ equally. The local inhibition and the global inhibition are processed by

$$\triangle \mathbf{P}^{\text{hdc}}_{\theta,h} = \sum_{i=1}^{n_\theta}\sum_{j=1}^{n_h} \mathbf{P}^{\text{hdc}}_{i,j} \psi^{\text{hdc}}_{u,v} - \varphi, \tag{15}$$

where $\boldsymbol{\psi}^{\text{hdc}}_{u,v}$ is the weight matrix for local inhibition. We limit all values in $\mathbf{P}^{\text{hdc}}$ to nonnegative values.

Activity normalization

The total activity in the multilayered head direction cells is kept at one by activity normalization. The normalized activity, $\mathbf{P}^{\text{hdc}'}_{\theta,h}$, is calculated by

$$\mathbf{P}^{\text{hdc}'}_{\theta,h} = \frac{\mathbf{P}^{\text{hdc}}_{\theta,h}}{\sum_{i=1}^{n_\theta}\sum_{j=1}^{n_h} \mathbf{P}^{\text{hdc}}_{i,j}}. \tag{16}$$

We describe the head direction update process in the multilayered head direction cell network according to the output of direction change and height change from 3D grid cell network in the following section.

Head direction update

The MD-CAN of the multilayered head direction cells performs the head direction update based on the direction change and height change by projecting activity from the current activated cells into nearby cells. The activity is shifted in yaw and height matrix according to the rotational direction change ω_θ and height change v_h, respectively. The activity injected is determined by two inputs. One is the activity of the sending unit, U^{hdc}. Another is the residue, η, which is calculated by fractional components of offsets, δ_{θ_f} and δ_{h_f}. The activity change $\triangle U^{\text{hdc}}_{lm}$ is calculated by

$$\Delta U^{\text{hdc}}_{lm} = \sum_{\theta=\delta_{\theta_0}}^{\delta_{\theta_0}+1}\sum_{h=\delta_{h_0}}^{\delta_{h_0}+1} \eta U^{\text{hdc}}_{(l+\theta)(m+h)}, \tag{17}$$

$$\begin{bmatrix} \delta_{\theta_0} \\ \delta_{h_0} \end{bmatrix} = \begin{bmatrix} \lfloor k_\theta \omega_\theta \rfloor \\ \lfloor k_h v_h \rfloor \end{bmatrix}, \tag{18}$$

$$\begin{bmatrix} \delta_{\theta_f} \\ \delta_{h_f} \end{bmatrix} = \begin{bmatrix} \lfloor k_\theta \omega_\theta - \delta_{\theta_0} \rfloor \\ \lfloor k_h v_h - \delta_{h_0} \rfloor \end{bmatrix}, \tag{19}$$

where k_θ and k_h are constants for head direction update. The residue matrix η is calculated by

$$\begin{aligned} \eta &= f(\delta_{\theta_f}, \theta - \delta_{\theta_0}) f(\delta_{h_f}, h - \delta_{h_0}), \\ f(a, b) &= \begin{cases} a, & \text{if } b = 1, \\ 1 - a, & \text{if } b = 0. \end{cases} \end{aligned} \tag{20}$$

4.3 Multilayered experience map

A multilayered experience map is a topological graphical map. It consists of many 3D spatial experiences, E_i. The transitions, T, connect relevant experiences. The experience creation is driven by the activity in the local view cells, the 3D grid cells and the multilayered head direction cells. We define each experience E_i by its associated conjunctive code of local view cell code V^{lv}_i, 3D grid cell code P^{gc}_i and multilayered head direction cell code P^{hdc}_i. The experience pose is $\mathbf{P}^{\text{exp}}_i$ which is estimated based on visual odometry. We define a

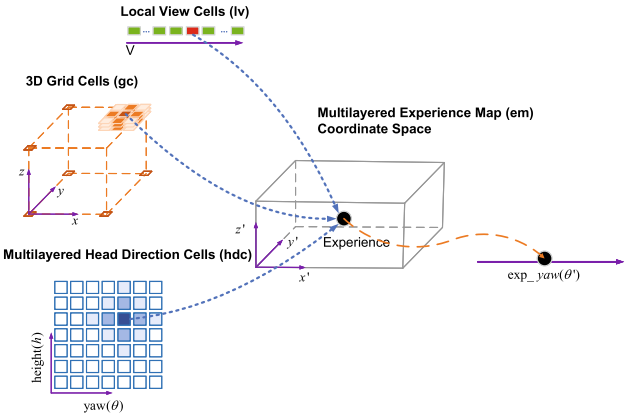

Fig. 4 The union coordinate space of the multilayered experience map. A 3D spatial experience E_i is encoded with certain local view cell V_i^{lv}, 3D grid cell P_i^{gc} and multilayered head direction cell P_i^{hdc}. The experience pose is estimated based on visual odometry. We create a new 3D spatial experience if the current conjunctive code is distinct from the existing conjunctive codes

3D spatial experience E_i as a tuple.

$$E_i = \{V_i^{\text{lv}}, P_i^{\text{gc}}, P_i^{\text{hdc}}, \mathbf{P}_i^{\text{exp}}\}. \tag{21}$$

Figure 4 shows an experience and its associated conjunctive code. The (x, y, z, θ) describes the 3D location within the 3D grid cells and the direction within the multilayered head direction cells associated with particular experience. Each local view cell in the V is associated with particular experience. Each spatial experience has a 3D spatial state (x', y', z', θ'). The union coordinate describes the location and direction of the experience in 3D space. We set an initial 4DoF pose (0, 0, 0, 0) for the first experience. The pose of experience is estimated based on the translational and rotational velocity. The algorithm of multilayered experience map creation and relaxation is described in Algorithm 4 (Online Appendix).

Multilayered experience map creation

If the existing 3D spatial experiences are insufficient for encoding the union pattern of the activity in the local view cells, the 3D grid cells and the multilayered head direction cells, we create a new 3D spatial experience. We use a score metric S_i to compare the current experience union code consisting of local view cell code V_i^{lv}, the 3D grid cell code P_i^{gc} and the multilayered head direction cell code P_i^{hdc} with all existing experiences' union codes consisting of the local view cell code V^{lv}, the 3D grid cell code P^{gc} and the multilayered head direction cell code P^{hdc}. The subscript i represents the index of the current experience.

$$S_i = \mu^{\text{lv}}|V_i^{\text{lv}} - V^{\text{lv}}| + \mu^{\text{gc}}|P_i^{\text{gc}} - P^{\text{gc}}| + \mu^{\text{hdc}}|P_i^{\text{hdc}} - P^{\text{hdc}}|, \tag{22}$$

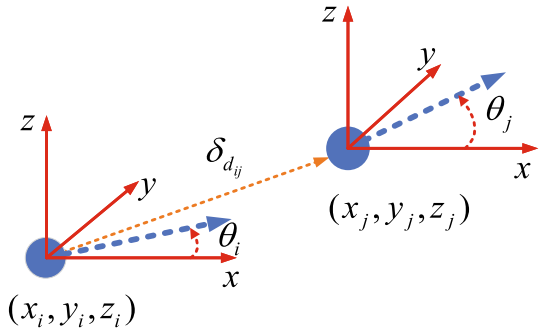

Fig. 5 A transition link from the 3D experience E_i to E_j. The 4DoF pose is indicated as the circles and blue dotted arrows

where μ^{lv}, μ^{gc}, μ^{hdc} weight each contribution of the local view cells, the 3D grid cells and the multilayered head direction cells, respectively. We create a new 3D spatial experience and its transition if the $\mathbf{S}$ for all existing 3D spatial experiences exceeds a value $S_{\max}$. We choose the 3D spatial experience associated with the lowest score as current active 3D spatial experience if all matching scores are less than the value. The active 3D spatial experience represents the current robot's 4DoF pose in 3D space.

The transition link stores the 4DoF pose change and the distance between one experience and another. The experience map correction and relaxation are performed using this transition information. A transition from E_i to E_j is shown in Fig. 5.

The transition T_{ij} stores the 4DoF pose change $\delta_{\mathbf{P}_{ij}^{\text{exp}}}$ and the distance $\delta_{d_{ij}}$.

$$T_{ij} = \{\delta_{\mathbf{P}_{ij}^{\text{exp}}}, \delta_{d_{ij}}\},$$
$$\delta_{\mathbf{P}_{ij}^{\text{exp}}} = \mathbf{P}_j^{\text{exp}} - \mathbf{P}_i^{\text{exp}} = \begin{bmatrix} x_j \\ y_j \\ z_j \\ \theta_j \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \\ z_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \delta_{x_{ij}} \\ \delta_{y_{ij}} \\ \delta_{z_{ij}} \\ \delta_{\theta_{ij}} \end{bmatrix}, \tag{23}$$

where the 4DoF pose $\mathbf{P}_j^{\text{exp}}$ of the new experience E_j is calculated according to the movement and previous 4DoF pose $\mathbf{P}_i^{\text{exp}}$. The new link T_{ij} between the two experiences is formed.

$$E_j = \left\{V_j^{\text{lv}}, P_j^{\text{gc}}, P_j^{\text{hdc}}, \mathbf{P}_i^{\text{exp}} + \delta_{\mathbf{P}_{ij}^{\text{exp}}}\right\}. \tag{24}$$

Multilayered experience map relaxation

When seeing a familiar scene, we inject visual activity into the local view cells, the 3D grid cells and the multilayered head direction cells associated with that scene. This causes that the robot's 4DoF pose is relocalized. Thus, the 4DoF pose of the robot changes from the new experience to an existing one. Meanwhile, the new transition link is learned. But the change in the relative 4DoF pose is different between the new

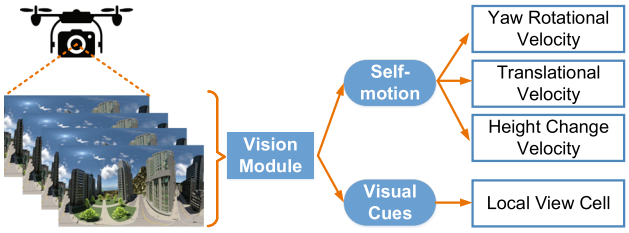

Fig. 6 Overview of components of the vision system. The output from vision system includes self-motion and visual cues which is as input of the 3D grid cell network and the multilayered head direction cell network for driving path integration

transition and the old transition due to the accumulated error of the 3D visual odometry.

In order to reduce the error of the relative 4DoF pose estimation, we utilize an iteration approach of the multilayered experience map correction and relaxation with an appropriate correction rate ψ according to the pose change information in the old transition and the new transition. The 4DoF pose change, $\delta_{\mathbf{P}_i}$, is calculated by:

$$\delta_{\mathbf{P}_i} = \psi \left[\sum_{k=1}^{N_t} (\mathbf{P}_k - \mathbf{P}_i - \delta^{\text{old}}_{\mathbf{P}_{ki}}) + \sum_{j=1}^{N_f} (\mathbf{P}_j - \mathbf{P}_i - \delta^{\text{old}}_{\mathbf{P}_{ij}}) \right], \tag{25}$$

where N_f, N_t are the amount of transitions from E_i to other 3D spatial experiences and from other 3D spatial experiences to E_i, respectively.

4.4 Vision module

The local visual cues and self-motion cues are provided by the vision system, as shown in Fig. 6. We extend the approaches in Milford and Wyeth (2008) for translational and rotational velocity estimation as well as visual template matching. The 3D visual odometry, aidvo, is based on average intensity difference by comparing a set of consecutive images. This process is not suitable for unrestricted movement in 6DoF but is adequate for the movement schemes presented here. We have also now implemented less restrictive visual odometry techniques including VINS-mono (visual inertial odometry) (Qin et al. 2018).

Image acquisition

Raw images were collected by a low-cost camera with low resolution. Grayscale images are resolution reduced and cropped into four regions for estimating yaw rotational velocity, translational velocity, height change velocity and visual template matching. We choose the regions and cropping size of sub-images according to the rules in Milford and Wyeth (2008) and Milford (2013). Patch normalization is used to improve the robustness of visual template matching. The

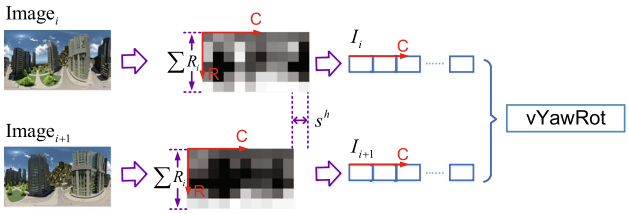

Fig. 7 Process for estimating yaw rotational velocity using consecutive intensity profiles

pixel intensities are calculated by:

$$I'_{xy} = \frac{I_{xy} - \mu_{xy}}{\delta_{xy}}, \tag{26}$$

where μ_{xy} is the mean. δ_{xy} is the standard deviation.

A scanline intensity profile is used in the vision processing approach, which is a one-dimensional vector formed from the sub-images. The 3D translational and rotational velocity are estimated using this profile. The visual template matching approach is also based upon the profile.

Estimating yaw rotational velocity

Yaw rotational velocity is calculated by comparing a set of consecutive images, as shown in Fig. 7. The two profiles I^i and I^{i+1} are shifted s^{h} in column dimension. Then, the average intensity difference between them is calculated.

$$d(I^i, I^{i+1}, s^{\text{h}}) = \frac{1}{w - |s^{\text{h}}|} \left(\sum_{n=1}^{w-|s^{\text{h}}|} \left| I^{i+1}_{n+\max(s^{\text{h}},0)} - I^{i}_{n-\min(s^{\text{h}},0)} \right| \right), \tag{27}$$

where s^{h} is the profile shift in column dimension. w is the width of the image.

$$s^{\text{h}}_{\text{m}} = \underset{s \in [\rho^h - w, w - \rho^h]}{\arg\min} \; d(I^i, I^{i+1}, s^{\text{h}}). \tag{28}$$

The rotational velocity $\Delta\theta$ is estimated by s^{h}_{m} multiplied by the constant σ^{h}. We can determine the σ^{h} empirically.

$$\Delta\theta = \sigma^{\text{h}} s^{\text{h}}_{\text{m}}. \tag{29}$$

Estimating translational velocity

Translational velocity is calculated by comparing profile difference in column dimension using a set of consecutive sub-images.

$$v = \min\left[\mu d(I^i, I^{i+1}, s^{\text{h}}), v_{\max} \right], \tag{30}$$

where the constant μ is determined empirically for scaling physical speed. In order to reduce the estimation error of

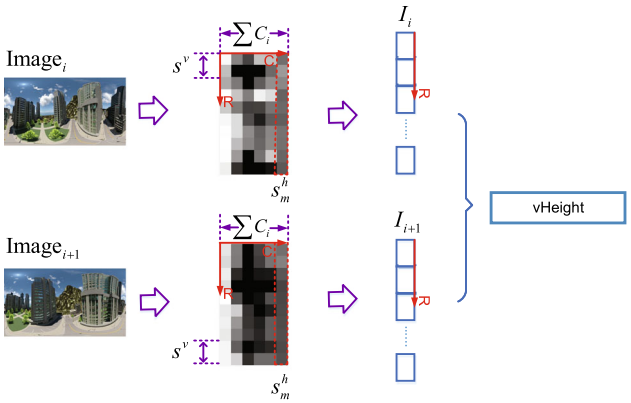

Fig. 8 Process for estimating height change velocity with consecutive intensity profiles

translational velocity, we set a maximum threshold $v_{\max}$ for filtering unusual values of profile difference.

Estimating height change velocity

The height change velocity is estimated based on a set of consecutive sub-images by comparing their profile difference in row dimension. Firstly, the current image is cropped with the offset $s^{\mathrm{h}}_{\mathrm{m}}$ at the best match of yaw estimate in order to reduce the effects of yaw rotation. As shown in Fig. 8, the red dotted rectangle area should be cropped.

The intensity difference $d()$ between I^i and I^{i+1} is calculated by

$$\begin{aligned} & d(I^i, I^{i+1}, s^h_m, s^v) \\ & = \frac{1}{h - |s^{\mathrm{v}}|}\left(\sum_{m=1}^{h-|s^{\mathrm{v}}|} \left| I^{i+1}_{m+\max(s^{\mathrm{v}},0)} - I^{i}_{m-\min(s^{\mathrm{v}},0)} \right| \right), \\ & I = \sum_{j=1}^{w-|s^{\mathrm{h}}_m|} I'_j, \end{aligned} \tag{31}$$

where s^h_m and s^v are offsets in column and row dimension, respectively. h is the height of the sub-image. w is the width of the sub-image. The d_{m} is the intensity difference in a set of consecutive images at the minimum offset.

$$\begin{aligned} & d_{\mathrm{m}} = \min_{\substack{s^{\mathrm{h}}_m \in [\rho^{\mathrm{h}} - w,\, w - \rho^{\mathrm{h}}] \\ s^{\mathrm{v}} \in [\rho^{\mathrm{v}} - h,\, h - \rho^{\mathrm{v}}]}} d(I^i, I^{i+1}, s^{\mathrm{h}}_m, s^{\mathrm{v}}), \\ & v_{\mathrm{h}} = \min[\mu d_{\mathrm{m}}, v^{\mathrm{h}}_{\max}], \end{aligned} \tag{32}$$

where the constant, μ , is calculated empirically for scaling physical speed.

Local view cell calculation

Each local view cell is paired with a visual template. When seeing a familiar scene, the local view cell associated with that view is activated. The visual templates are calculated and stored using the scanline intensity profile, as shown in Fig. 9. We set a threshold to control the generation of the templates. If the profile difference is less than the threshold, the current profile is matched to the template. Otherwise, a new template is created. For more details about visual template matching algorithm, see Milford and Wyeth (2008).

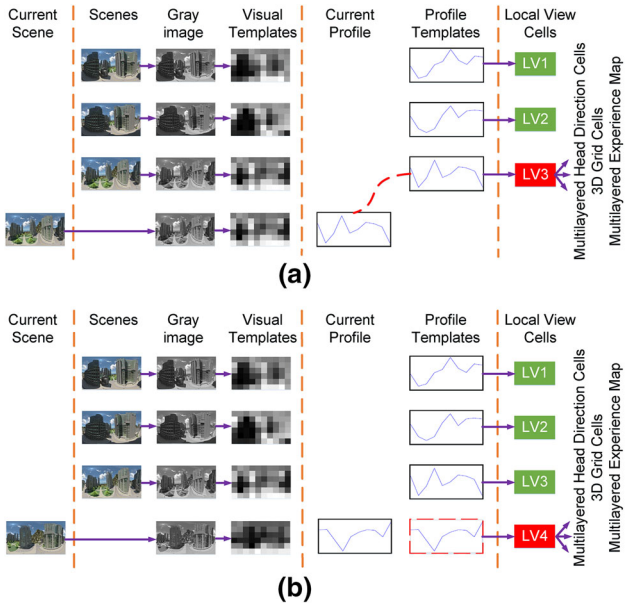

Fig. 9 The local view cell calculation. **a** The LV3 is activated when the current profile is matched to its associated profile template. **b** A new profile template and a new local view cell LV4 are created and activated if there is no profile matched

5 Experimental setup

In this section, we describe the four groups of datasets and the experimental parameters for evaluating the NeuroSLAM performance. To evaluate the system performance in various conditions, we constructed two groups of synthetic datasets, named SynPerData and SynPanData, using perspective camera and panoramic camera, respectively, in synthetic 3D urban environments. We also collected a group of real-world datasets in a 3D carpark, named QUTCarparkData, for evaluating the system in read-world environments. In addition, in order to demonstrate 3D mapping performance, we use one of maplab datasets (cla-floor-f) (Schneider et al. 2018). This dataset was collected in a multi-level building. It contains consecutive images collected by monocular camera and IMU sensor data. The properties of the datasets are given in Table 4. The datasets are available at https://github.com/cognav/NeuroSLAM/blob/master/Datasets. The key parameters of each module are described in the following. At the end of this section, we analyzed the effects of these parameters on the system performance.

Table 4 Four groups of datasets in various conditions

Datasets	Environment	Camera	Viewpoint	Number	Resolution	Purpose
SynPerData	Synthetic 3D urban environment	Perspective camera	Same	5000	160×120	To evaluate 3D localization and mapping performance with a monocular camera with the same viewpoint
SynPanData	Synthetic 3D urban environment	Panoramic camera	Opposite	3000	480×200	To evaluate 3D localization and mapping performance with a panoramic camera with the opposite viewpoint
QUTCarparkData	Real-world indoor and outdoor 3D environment	Perspective camera	Same	12583	480×270	To evaluate 3D localization and mapping performance in a real-world indoor and outdoor 3D environment
cla-floor-f	Real-world indoor multi-level building	Perspective camera	Same	3479	752×480	To evaluate 3D mapping performance in a real-world indoor 3D environment based on visual inertial odometry

5.1 Synthetic datasets acquisition

To generate photorealistic synthetic images, we used the Cycles raytracing engine implemented in Blender. Furthermore, in order to get wide-angle images, we used a panoramic camera model implemented in Blender by Zhang et al. (2016). We generated two groups of datasets where the camera traversed at least two circles across trajectories with the same or opposite (direction) viewpoint in 3D urban environments. The SynPerData was produced with a perspective camera model for testing the system performance with monocular camera. The SynPanData was used for testing the system performance with a panoramic camera. The snapshots of 3D trajectory and scenes are shown in Fig. 10. All datasets include 8000 consecutive scene images. We extracted the 4DoF poses of camera from the ground truth trajectories for evaluating the performance of mapping.

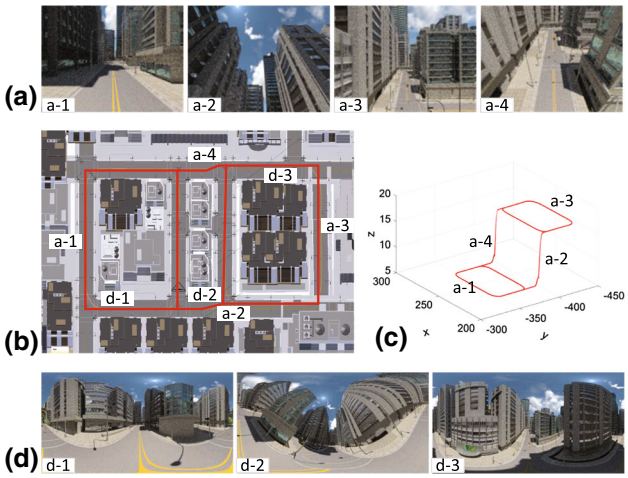

Fig. 10 Snapshots of the synthetic 3D urban environments. **b** and **c** show the camera trajectory in the SynPerData. **a** is some key snapshots in the SynPerData when the perspective camera moves through the path in plane and vertical (up and down) space. **d** is some key snapshots in the SynPanData when the panoramic camera moves through the synthetic 3D space. The synthetic 3D urban environment consists of buildings with various scene materials, trees, roads, transportation facilities, etc. The light shade is changing in part of the roads

5.2 Real-world dataset acquisition

In order to evaluate the NeuroSLAM performance in a real-world 3D environment, we collected the QUTCarParkData over a two-level carpark consisting of indoor and outdoor parts on a university campus, as shown in Fig. 11. A smooth slope connects these two levels. We gathered the image dataset using a commodity smartphone camera in the carpark. The smartphone was deployed in the front of the bike. We drove the bike through the car park for over two laps. The total distance traversed was approximately 600 m. There are 12583 480 $\times$ 270 pixel images, comprising about 2.1 GB of imagery.

5.3 Experimental parameter setting

Tables 5, 6, 7, 8 and 9 show the parameters and their values in the NeuroSLAM system. The principles and rules of tuning parameters in our system are based on Ball et al. (2013).

The NeuroSLAM system performance is dependent on the tuning of several parameters of the local view cells, the 3D grid cells and the multilayered head direction cells. The energy input of visual calibration and the tuning of inhibition for these conjunctive cells also have an effect on 3D mapping performance. If the energy input from familiar visual

Table 5 3D grid cell parameters

Parameter	Value		
	SynPerData	SynPanData	QUTCarparkData
n_x, n_y, n_z	36 × 36 × 36	36 × 36 × 36	36 × 36 × 36
Cell size	1 ×1 × 1	1 ×1 × 1	1m ×1m × 1m
N_{excit}	7 ×7 × 7 cells	7 ×7 × 7 cells	7 ×7 × 7 cells
N_{inhib}	5 ×5 × 5 cells	5 ×5 × 5 cells	5 ×5 × 5 cells
$E_{\text{var}}(x, y, z)$	1.5, 1.5, 1.5	1.5, 1.5, 1.5	1.5, 1.5, 1.5
$I_{\text{var}}(x, y, z)$	2, 2, 2	2, 2, 2	2, 2, 2
φ	0.0002	0.00025	0.0002
δ	0.1	0.1	0.1

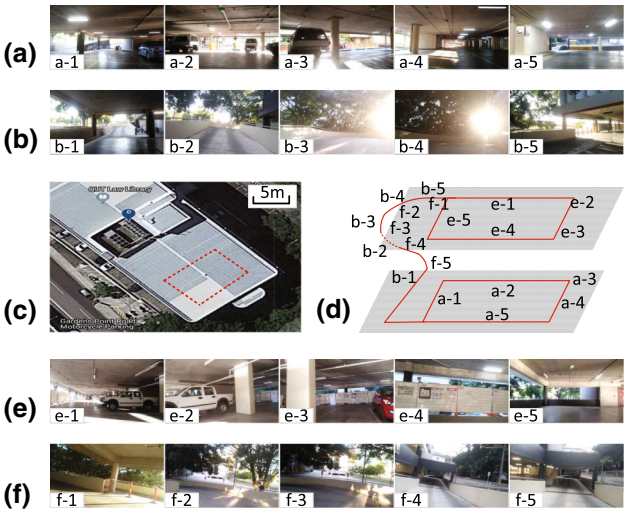

Fig. 11 Snapshots of the real-world 3D university campus carpark in QUTCarparkData. **c** and **d** show the topographic map and the 3D trajectory. **a** and **e** are snapshots in level one and two, respectively. **b** and **f** are snapshots when moving up and down through the slope connecting these two levels. We collected these data at 3:00 PM. The sunlight was strong in some regions, e.g., (b-3) and (b-4)

Table 6 Multilayered head direction cell parameters

Parameter	Value		
	SynPerData	SynPanData	QUTCarparkData
$n_\theta, n_{\text{height}}$	36 × 36	36 × 36	36 × 36
Cell size	10 deg × 1	10 deg × 1	10 deg × 1m
N_{excit}	8 × 8 cells	8 × 8 cells	8 × 8 cells
N_{inhib}	5 × 5 cells	5 × 5 cells	5 × 5 cells
$E_{\text{var}}(\theta, h)$	1.9, 1.9	1.9, 1.9	1.9, 1.9
$I_{\text{var}}(\theta, h)$	3.0, 3.0	3.1, 3.1	3.1, 3.1
φ	0.0002	0.0003	0.0002
δ	0.001	0.01	0.1

Table 7 Local view cell parameters

Parameter	Value		
	SynPerData	SynPanData	QUTCarparkData
d_m	0.15	3.12	3.4
s_x	5	5	5
s_y	3	3	3
$vt_{\text{panoramic}}$	0	1	0
vt_{step}	1	2	5

Table 8 Multilayered experience map parameters

Parameter	Value		
	SynPerData	SynPanData	QUTCarparkData
$S_{\max}$	40	20	30
η	0.5	0.005	0.5
Loops	5	1	5

Table 9 Visual odometry parameters

Parameter	Value		
	SynPerData	SynPanData	QUTCarparkData
$R_{x,y}^{V_{\text{trans}}}$	16 : 145	181 : 300	1 : 480
	31 : 90	51 : 150	1 : 270
$R_{x,y}^{V_{\text{height}}}$	11 : 150	181 : 300	1 : 480
	11 : 110	51 : 150	1 : 270
$R_{x,y}^{V_{\text{rot}}}$	16 : 145	181 : 300	1 : 480
	31 : 90	51 : 150	1 : 480
$V_{\max}^{\text{trans}}$	0.5	0.22	0.4
$V_{\max}^{\text{height}}$	0.45	0.3	0.4
$V_{\max}^{\text{rot}}$	2.5	10	4.2
s_x	26	36	30
s_y	20	20	30
FOV_x	81.5	90	75
FOV_y	70	50	60
ODO_{step}	1	2	5

calibration or the value of inhibition is too high, the attractor network dynamics may be unstable. In our experiments, the values of excitation, inhibition and visual calibration are given in Tables 5 and 6. These parameters enable the system operating robustly in various environments.

In addition, the maximum visual template distance threshold, d_m, is essential to affect the performance of visual template matching. If the threshold is too high, it will cause too many false visual template matches. The visual template parameters are given in Table 7.

The 3D mapping performance is also affected by some other parameters. An appropriate correction rate is important for reliable loop closure in the multilayered experience map correction process. The experience threshold is essential to affect experience matching performance. The experience map parameters are given in Table 8. In addition, the maximum yaw rotational velocity, translational velocity and height change velocity were used to reduce the error of velocity estimation. Table 9 shows the 3D visual odometry parameters.

In the experiments based on the cla-floor-f dataset, most of the key parameters of the 3D grid cell network, the multilayered head direction cell network, the local view cells and the multilayered experience map are same as the parameters based on the SynPerData except the threshold of local view and experience map. The d_m of local view cell is 0.28. The $S_{\max}$ of the multilayered experience map is 40.

5.4 Trajectory evaluation metrics

In order to evaluate the mapping performance of our methods quantitatively, we use absolute trajectory error (ATE) and relative error (RE) as metrics by computing root-mean-square error (RMSE) (Zhang and Scaramuzza 2018). They are commonly used to evaluate the accuracy of odometry or SLAM. The ATE can give a single number metric for the position or rotation estimation. However, the RE can measure the relative error between sub-trajectories. Therefore, we can evaluate the SLAM quality from different aspects by combining the ATE and the RE.

We need to align and match different estimated trajectories with the ground truth trajectory before computing translation errors. However, trajectory alignment and pose node matching are difficult because there are different numbers of pose nodes in trajectories generated by several types of SLAM systems. These pose nodes are not evenly distributed with respect to distance travelled. In order to build pose correspondence between the estimated trajectory and the ground truth trajectory correctly, we utilize an approach to key position matching. As shown in Fig. S1 (Online Appendix), we extract specified number of key positions randomly at the turns in order. Then, the mean of each group of key positions is calculated. Finally, we calculate the ATE and the RE based on the mean of each group of key positions. Additionally, we evaluate the accuracy of each pair of key positions by computing translation error.

The ATE of key positions is estimated by

$$\mathrm{ATE}_{\mathrm{pos}} = \left(\frac{1}{n} \sum_{\mathrm{i}=1}^{n} (\triangle p_i)^2 \right)^{\frac{1}{2}}, \tag{33}$$

where $\triangle p_i$ represents the difference of 3D distance.

The RE of key positions is estimated by

$$\mathrm{RE}_{\mathrm{pos}} = \left(\frac{1}{m-k+1} \sum_{\mathrm{i}=\mathrm{k}}^{m} (\triangle p_i)^2 \right)^{\frac{1}{2}}. \tag{34}$$

6 Results

In this section, we present the overall 3D mapping results and the performance of each module of our system including multilayered experience map, local view cell activity, active experiences, 3D grid cell activity, multilayered head direction cell activity and visual odometry. In addition, we provide the quantitative evaluation by comparing NeuroSLAM with the state-of-the-art 3D visual SLAM systems, ORB-SLAM (Mur-Artal and Tardós 2017) and LDSO (Gao et al. 2018). Finally, we show the mapping results by integrating NeuroSLAM with VINS-mono (Qin et al. 2018), which is a visual inertial odometry based on the monocular camera. The videos of the experimental results are available at https://github.com/cognav/NeuroSLAM.

6.1 NeuroSLAM results

Firstly, we show the topologically correct 3D experience maps produced by the NeuroSLAM system with the three groups of datasets including SynPerData, SynPanData and QUTCarparkData. The quantitative comparison between experience map, odometry map and ground truth map for each group of experiment is presented. We also analyze the learning and recall results of visual templates and experiences. Then, some exemplar snapshots of activity packets in the 3D grid cell network and the multilayered head direction cell network are presented for demonstrating the operational process of the networks. At the end of this section, we show some key snapshots of the 3D visual odometry process.

6.1.1 Multilayered experience map

NeuroSLAM, like RatSLAM, does not generate a strictly metric 3D Cartesian spatial map. In contrast, it creates a topologically consistent 3D spatial representation with some

locally metric information. Consequently, the typical 3D mapping performance indicator is topological consistency rather than geometric accuracy. In order to first analyze the topology consistency of the experience map on a global scale, we have adjusted the scale of experience maps according to the ground truth map without changing their topology.

The following shows three groups of multilayered experience maps created by the NeuroSLAM system based on the SynPerData, SynPanData and QUTCarparkData. The 3D view and top view of each group of map are shown in the following figures. By comparing the odometry map, experience map and ground truth map, the topological consistency is shown more clearly. In addition, we quantitatively analyzed the root-mean-square error (RMSE) of the map by the absolute trajectory error (ATE) and the relative error (RE) introduced in section 5.4.

3D mapping results based on the SynPerData

As shown in Fig. 12, the correct and consistent map topology is revealed by comparing the multilayered experience map and the odometry map with the ground truth map. The overall topology in the multilayered experience map is consistent with the ground truth map. After the robot comes back to the familiar lower level from the upper level after completing several loops, the topology is still consistent with the ground truth map due to loop closure and map relaxation processes. However, the topological error in the odometry map increases with the distance traversed, due to the odometric drift error, which is shown in Fig. 12b.

Though the key performance indicator of NeuroSLAM is topological consistency, we add several new quantitative analyses of geometric accuracy for providing more informative evaluation. We provide the quantitative evaluation by computing the ATE and RE. Firstly, we extract 19 groups of key positions at the turns in order. Each group consists of five key positions through five trials randomly. The mean of each group of key positions is estimated, as shown in Fig. S2 (Online Appendix). We can see from Fig. S2 (Online Appendix) the majority of the key positions in the NeuroSLAM trajectory are near to the key positions in the ground truth trajectory.

Figure 13 shows the translation error of each group of key positions between the multilayered experience map, the odometry map and the ground truth map, respectively. Both the experience map error and the odometry map error are similar in some key positions because the experience pose is estimated based on the odometry. The translation error of the aidvo becomes larger with the increasing distance traversed due to the accumulate error of odometry. However, NeuroSLAM is more accurate in the key position set from 15 to 20 because of loop closure and map relaxation processing when revisiting familiar scenes. The quantitative results are consistent with the qualitative results in Fig. 12 and Fig. S2 (Online Appendix).

In order to evaluate the accuracy in overall trajectory, we estimate the ATE of each set of key positions between the NeuroSLAM map, the aidvo map and the ground truth map, respectively, based on the SynPerData, as shown in Fig. 14a. Each box represents the RMSE in a set of key positions, which indicates the translation error through each trajectory. NeuroSLAM has higher accuracy than the aidvo in the majority of key positions. In the set (1–16) and the set (1–19), the median of NeuroSLAM is also lower than that of the aidvo. The results are consistent with Fig. S2 (Online Appendix) and Fig. 13.

In addition, we evaluate the RE of each sub-trajectory between the estimated map and the ground truth map based on the SynPerData. Figure 14b shows the RE of key positions in each sub-trajectory between the NeuroSLAM map, the aidvo map and the ground truth map, respectively. The NeuroSLAM system has better performance than the aidvo in the majority of the key positions.

3D mapping results based on the SynPanData

As shown in Fig. 15, the 3D mapping results are presented based on the SynPanData. The topology in the multilayered experience map is consistent with the ground truth map. The topological error is smaller in the lower level than in the upper level. The main reason is that the drift error from visual odometry becomes large in the visually sky-dominated regions; the scenes captured by the panoramas camera change minimally with movement. Thus, the velocity estimation is not accurate. The loop closure and relaxation in the multilayered experience map work well when revisiting the familiar scenes from opposing viewpoints, as shown in Fig. 15a, b.

In addition, we also provide the quantitative evaluation by computing the ATE and the RE based on the SynPanData. Firstly, we extract 19 groups of key positions at the turns in order. Each group consists of five key positions through five trials randomly. The mean of each group of key positions is estimated, as shown in Fig. S3 (Online Appendix). We can see from Fig. S3 (Online Appendix) the majority of the key positions in the NeuroSLAM trajectory are near to the key positions in the ground truth trajectory.

Figure 16 shows the translation error of each group of key positions between the multilayered experience map, the odometry map and the ground truth map, respectively, based on the SynPanData. The translation error of the aidvo increases with the distance traversed due to the accumulate error of odometry. However, the error of the NeuroSLAM trajectory in several key positions is small because of loop closure and map relaxation processing when revisiting familiar scenes. The quantitative results are consistent with the qualitative results in Fig. 15 and Fig. S3 (Online Appendix).

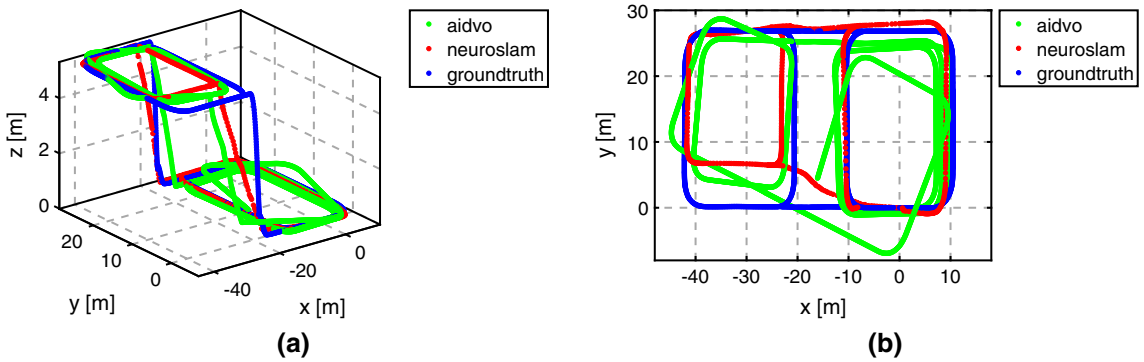

Fig. 12 The NeuroSLAM map (neuroslam), the aidvo map (aidvo) and the ground truth map (groundtruth) based on the SynPerData. **a** 3D view; **b** top view. Loop closure and map relaxation process in the multilayered experience map overcome the odometric drift error

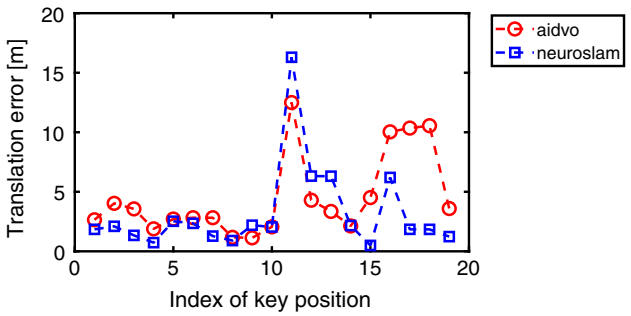

Fig. 13 The translation error of each group of key positions between the NeuroSLAM map, the aidvo map and the ground truth map, respectively, based on the SynPerData

Figure 17a shows the ATE of each set of key positions between the NeuroSLAM map, the aidvo map and the ground truth map, respectively, based on the SynPanData. Each box represents the RMSE in a set of key positions, which indicates the translation error through each trajectory. NeuroSLAM has similar accuracy to the aidvo in the majority of key positions. Moreover, in the set (1–16) and the set (1–19), NeuroSLAM has better performance than the aidvo due to map relaxation processing when revisiting familiar scenes. The results are consistent with Fig. S3 (Online Appendix) and Fig. 16.

Figure 17b shows the RE of key positions in each sub-trajectory between the NeuroSLAM map, the aidvo map and the ground truth map, respectively, based on the SynPanData. NeuroSLAM has better performance than the aidvo in the majority of the key positions.

3D mapping results based on the QUTCarparkData

As shown in Fig. 18, the 3D mapping results based on the QUTCarparkData are presented. We can see from Fig. 18a, b the overall topology is consistent with the floor plan as shown in Fig. 11. Due to a lack of accurate ground truth trajectory for this particular experiment, we did not provide a quantitative evaluation in this case. Though the visual odometry has drift errors, the multilayered experience map retains a consistent topology using loop closure and map relaxation when revisiting familiar scenes. As shown in Fig. 18a, b, when the

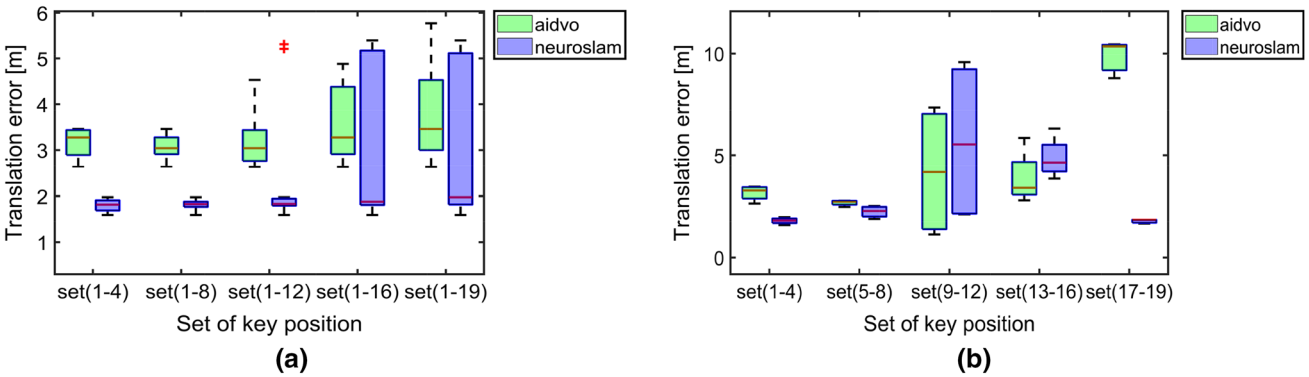

Fig. 14 The ATE and RE of each set of key positions between the NeuroSLAM map, the aidvo map and the ground truth map, respectively, based on the SynPerData which are shown as a series of boxplots. **a** The ATE; **b** the RE. Each box represents the RMSE in a set of key positions. For example, the box of the space (1–4) includes key position sets of {1}, {1, 2}, {1, 2, 3} and {1, 2, 3, 4}. The box in the middle indicates the two quartiles of estimation errors, the line through the box the median and the whiskers the upper and lower quartiles

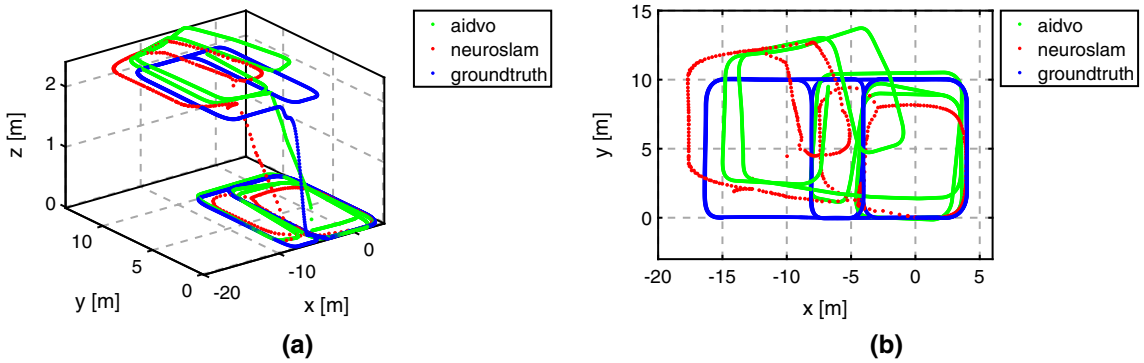

Fig. 15 The NeuroSLAM map, the aidvo map and the ground truth map based on the SynPanData. **a** 3D view; **b** top view. Loop closure and map relaxation processes in the multilayered experience map overcome the drift error of odometry

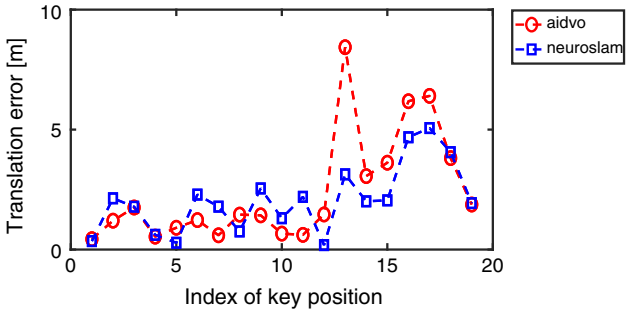

Fig. 16 The translation error of each group of key positions between the NeuroSLAM map, the aidvo map and the ground truth map, respectively, based on the SynPanData

robot comes back to the familiar places at the lower level from the upper level after traversing several loops, the recall of familiar experiences enables the robot to be relocated to the correct position.

Overall, the results of these three groups of experiments have shown the topological consistency and geometric accuracy of the multilayered experience maps. Though these up-level maps have a slight offset due to the long-term odometry drift, the layout of the road network is very close to the ground truth map. When the robot revisited a familiar place again, loop closure occurred and the experience map correction and relaxation occurred continually. The map correction process worked well during loop closure. The visual odometry map without loop closure has slight drift over the long term due to the cumulative errors of odometry. After sufficient map relaxation cycles, the multilayered experience map reached a completely stable configuration.

6.1.2 Local view cell activity

Figure 19 indicates visual templates generated with the three groups of datasets. The dotted line shows the start frame and the end frame during loop closure. Periods of no new template additions indicate times when the robot was moving through already learned environments. The performance of visual place recognition is good with high recognition rates, especially with SynPerData, as shown in Fig. 19a. Specific to Fig. 19b, the segments of decreasing template IDs show the robot traversed along a previously learned path in the opposite direction due to the panoramic matching. The performance of visual place recognition in real world is also good as shown in Fig. 19c.

6.1.3 Active experiences

Figure 20 shows active experiences were learned or recalled when moving through a novel or familiar environments, respectively. The numbers of experiences in three groups of maps are very close to the numbers of visual templates as shown in Fig. 20. In Fig. 20a–c, the numbers of experiences learned with the balanced threshold of experience creation are suitable to create a high-quality experience map.

6.1.4 3D grid cell activity

The individual 3D grid cells fired at multiple distinct locations in the 3D environments. The wrapping connectivity of the 3D grid cell network encodes the large size of the 3D space by reusing cells. Figure 21 shows some snapshots of active 3D grid cells packet when moving through planar and vertical space. The active packet moved from the initial position to the right and then wrapped around to the left in the plane, as shown in Fig. 21 from (a-1) to (a-5). In Fig. 21b, c, the cells in various vertical areas were activated when the camera moved in vertical space.

Figure 22 shows the history of the most active cells during mapping with the three groups of datasets. Many cells in two planes were activated repeatedly due to the robot moving in the 3D environment for a long duration. Some cells encoded the 3D space repeatedly. Part of the cells in different heights was activated when the robot moved in vertical space, as shown in Fig. 22a, b. Due to the variations in spatial scale

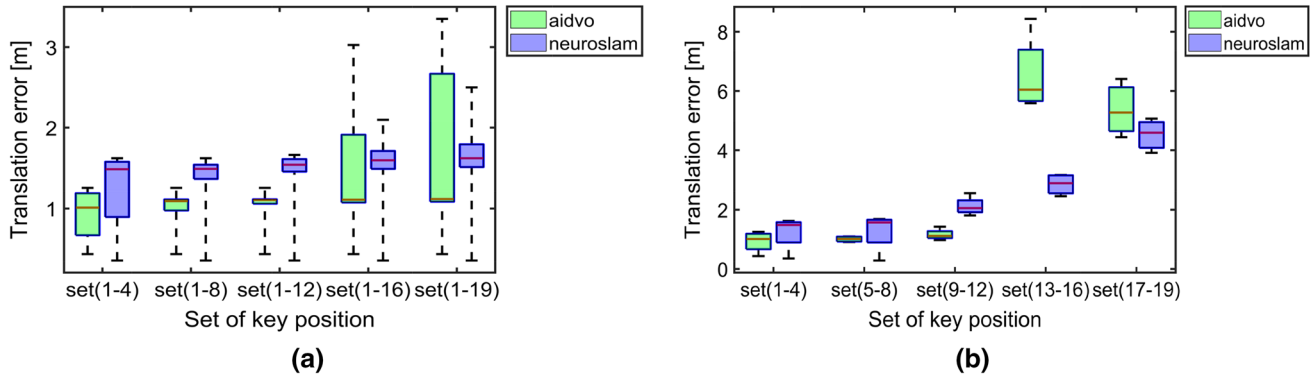

Fig. 17 The ATE and RE of each set of key positions between the NeuroSLAM map, the aidvo map and the ground truth map, respectively, based on the SynPanData. **a** The ATE; **b** the RE. Each box represents the RMSE in a set of key positions, e.g., the box of the space (1–4) including key position sets of {1}, {1, 2}, {1, 2, 3} and {1, 2, 3, 4}. The box in the middle indicates the two quartiles of estimation errors, the line through the box the median and the whiskers the upper and lower quartiles

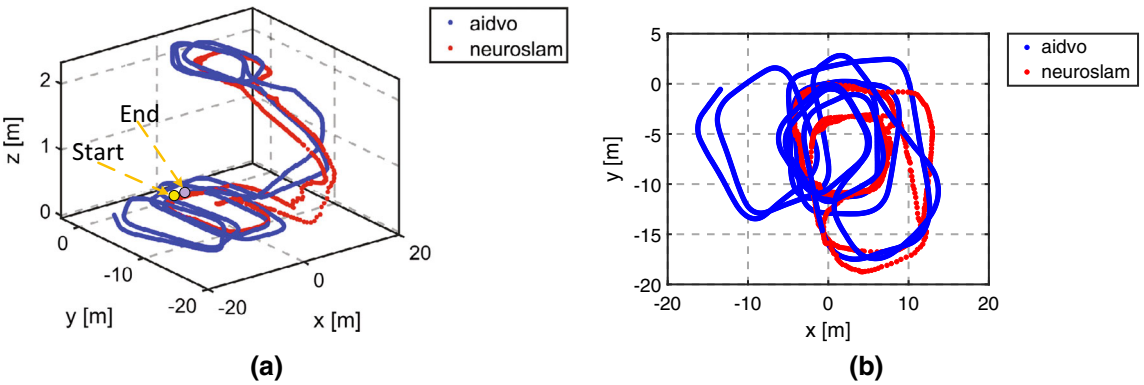

Fig. 18 The NeuroSLAM map, the aidvo map and the ground truth map based on the QUTCarparkData. **a** 3D view; **b** top view. Loop closure and multilayered experience map relaxation processes overcome the odometric drift error

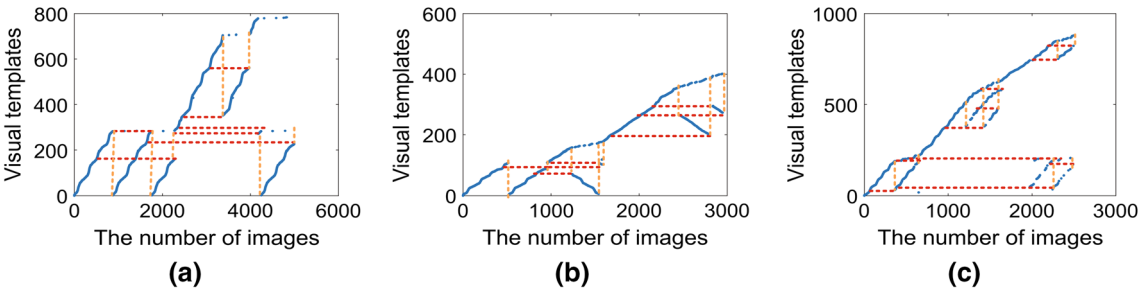

Fig. 19 Visual template learning and recall. The history of templates learned and recalled with SynPerData, SynPanData and QUTCarparkData is shown in (**a**), (**b**) and (**c**), respectively. The vertical axis shows the history of visual template ID. The horizontal axis shows the image index in sequence of images. The dotted line shows the start frame and the end frame during loop closure. Periods of no new template additions indicate times when the robot was moving through already learned environments and previous visual templates are recalled. Both the index of the visual templates and the index of images start from 1

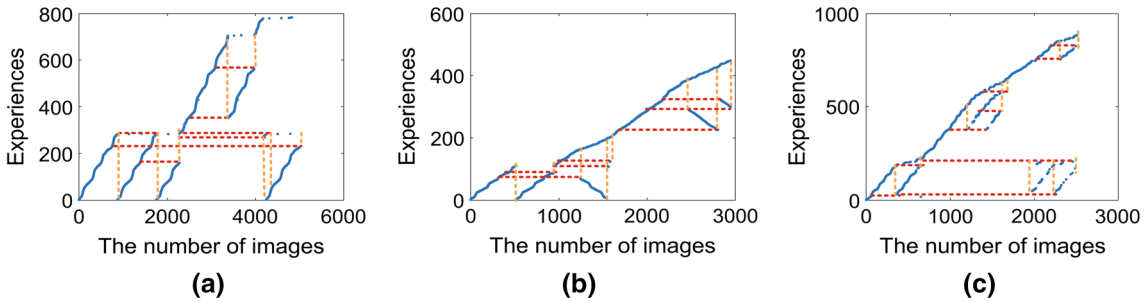

Fig. 20 Experience learning and recall. **a–c** show the history of experience learning and recall with SynPerData, SynPanData and QUTCarparkData, respectively. The vertical axis demonstrates the experience ID. The horizontal axis demonstrates the image index in sequence of images. The dotted line shows the start frame and the end frame during loop closure. Periods of no new experience additions indicate times when the robot was moving through already learned 3D environments. Stored experiences were activated and recalled if the current conjunctive code was similar to the stored codes

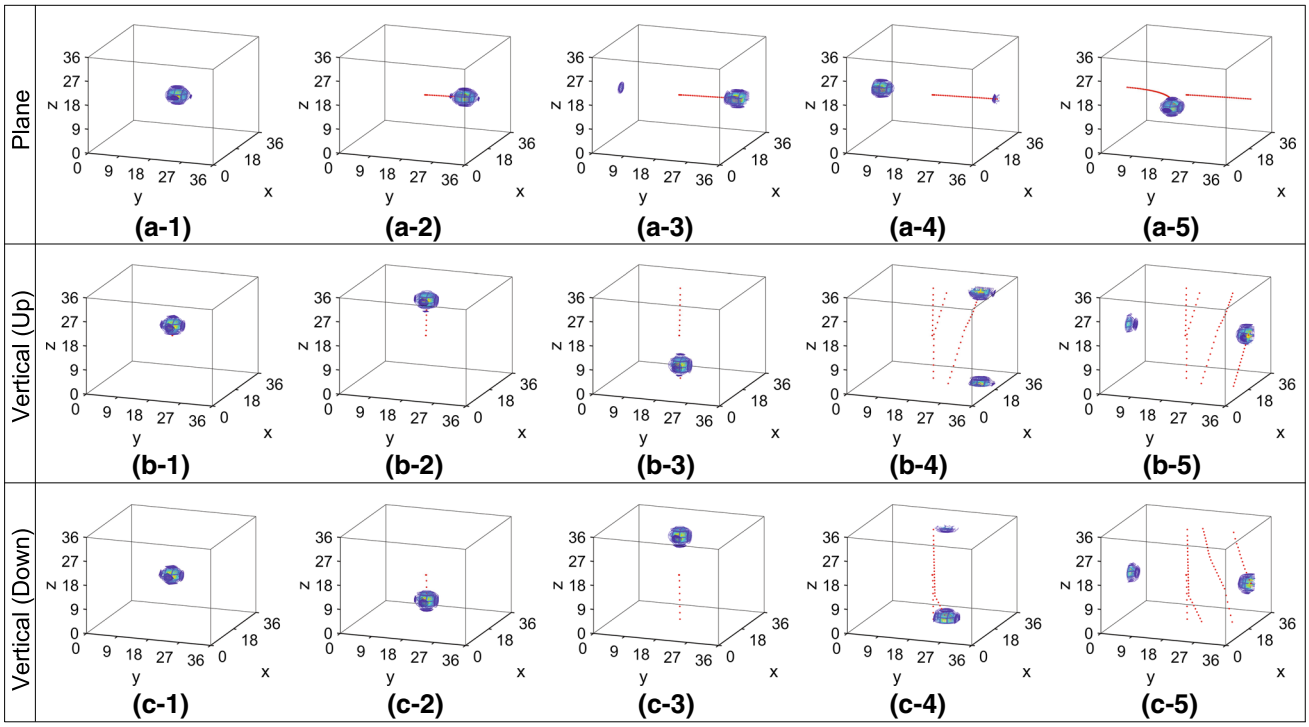

Fig. 21 Snapshots of 3D grid cell packets activated when the robot was moving through a plane or vertical space. (a-1) to (a-5) are some key snapshots when a packet moved through a plane from center to right and then wrapped around from the right boundary to the left boundary. (b-1) to (b-5) show a packet moving through vertical space from the center to the upper network boundary and then wrapping around to the bottom boundary when the camera was moving up on the slope shown in Fig. 11. (c-1) to (c-5) show a packet moved through from center to bottom boundary and then wrapping around to up boundary when the robot was moving down on the slope shown in Fig. 11

and size of the different testing environments, the activated cells in Fig. 22c are relatively sparse compared with the other two groups.

6.1.5 Multilayered head direction cell activity

Figure 23 shows some snapshots of active multilayered head direction cells. When the robot moved in a specific planar space, the height was relatively stable, and active cells were shifted across the yaw dimension, as shown in Fig. 23 from (a-1) to (a-5). The cells at opposing boundaries with wrapping connections can be activated if the accumulated angle is more than 360° or less than −360°. In Fig. 23b, c, the cells in different heights were activated when the robot moved through the vertical space. The packet was shifted in height dimension.

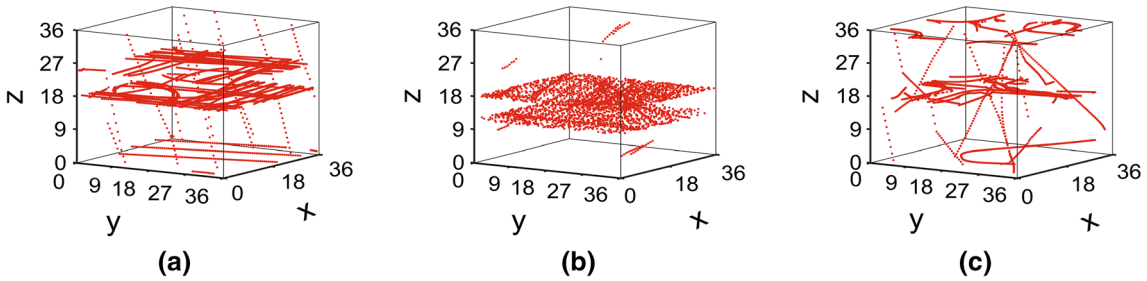

Fig. 22 The history of the most active 3D grid cells. **a–c** show the history of the most active grid cells when the NeuroSLAM system mapped with SynPerData, SynPanData and QUTCarparkData, respectively. The grid cells in the plane and vertical dimensions were activated when the robot was moving through the 3D space. Many cells were activated repeatedly as shown by the dense dots

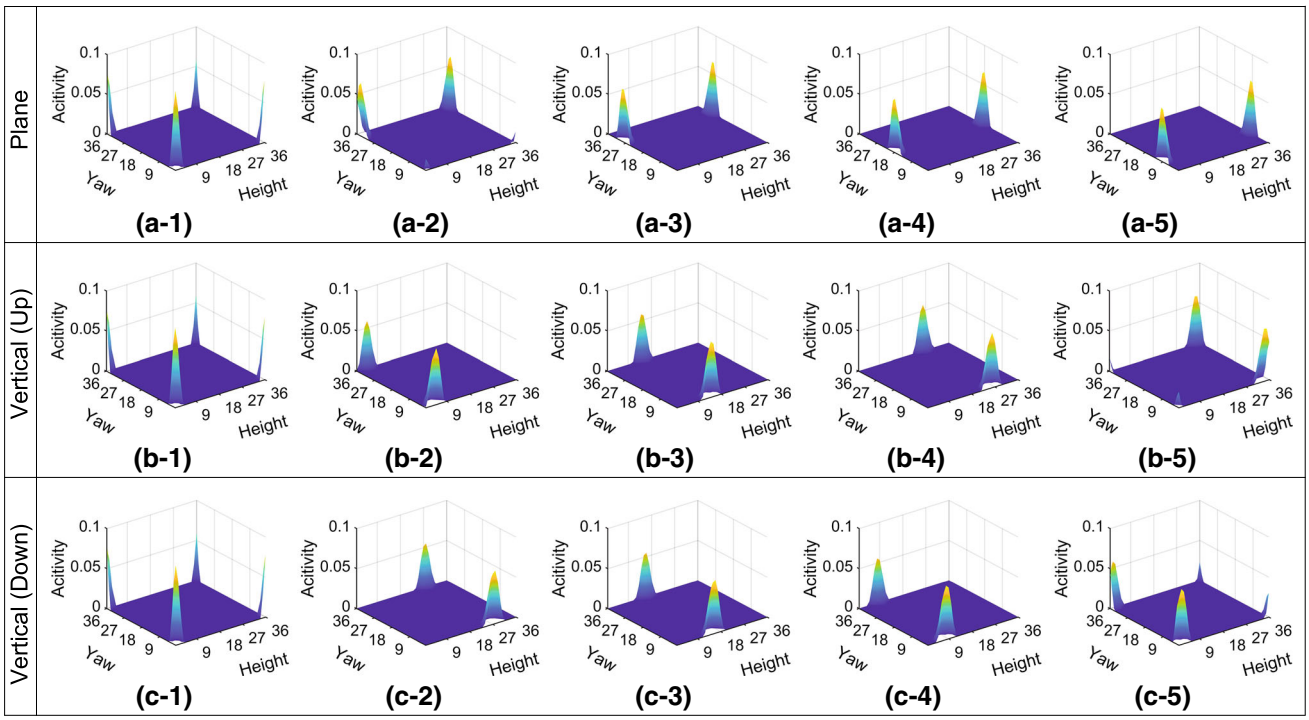

Fig. 23 Snapshots of active multilayered head direction cells packet. (a-1) to (a-5) are some key snapshots of a packet wrapping from the boundary of 0 to 36 and then moving across yaw dimension when the robot rotated clockwise. (b-1) to (b-5) are some key snapshots of a packet moving across the height dimension from 0 to 36 when the robot moved up on a slope without changing its head direction. (c-1) to (c-5) are some key snapshots of a packet moving across the height dimension from 36 to 0 when the robot moved down on a slope without changing its head direction

Figure 24 shows the history of the most active multilayered head direction cells. Many cells in various heights were activated when the robot moved in these planes, as shown by the horizontal line in Fig. 24. Some cells in different heights were activated when the camera moved vertically, as shown by the vertical line in Fig. 24. Due to the variations in spatial scale and size of the vertical spaces, the activated cells at different heights in Fig. 24c are relative dense compared with the other two groups.

6.1.6 Visual odometry

Figure S4 (Online Appendix) shows some key snapshots of visual odometry for estimating translational and rotational velocity during the first lap at the stages of 0°, − 90°, − 180°, − 270°, − 360°, up and down. These velocities were estimated based on the SynPerData. The translational velocity was not stable due to sudden changes between successive scenes, as shown in Fig. S4 (Online Appendix) from (a-1) to (a-7). We set the max velocity threshold, 0.2, to reduce the effects of scene change suddenly, e.g., sunlight. The vision

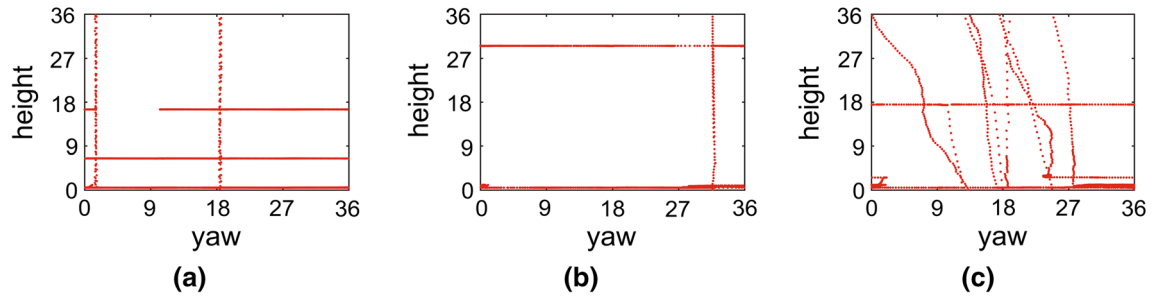

Fig. 24 The history of the most active multilayered head direction cells. **a–c** are the most active multilayered head direction cells when the NeuroSLAM system mapped with SynPerData, SynPanData and QUTCarparkData, respectively. Many head direction cells were activated in various of height dimensions which represented the robot's direction in different height of the vertical space

system can estimate height change velocity when moving up and down in vertical space, as shown in Fig. S4 (Online Appendix) (b-6) and (b-7). The rotational velocity was estimated stably, as shown in Fig. S4 (Online Appendix) from (c-1) and (c-7). Figure S4 (Online Appendix) (d) and (e) shows the accumulated angle and map based on visual odometry.

6.2 Comparison with state-of-the-art 3D SLAM

In order to evaluate the quality and topological consistency of the NeuroSLAM system, we compare our method with two types of state-of-the-art conventional 3D visual SLAM, ORB-SLAM (Mur-Artal and Tardós 2017) and LDSO (Gao et al. 2018) based on the SynPerData. ORB-SLAM is a feature-based monocular visual SLAM system, which is commonly used for comparison. LDSO is a monocular visual SLAM system based on direct sparse visual odometry (DSO), in which the loop closure is implemented based on feature matching.

All 3D maps generated from these methods are shown in Fig. 25. The topology in these maps is consistent with the ground truth map. However, the map generated from LDSO has large errors due to incorrect estimation during motion up or down the slope. If the motion of the camera changes quickly in up and down direction, LDSO has large errors. Thus, the results from LDSO have drift error in several subtrajectories. We can see from Fig. 25a that the NeuroSLAM system can generate a map with consistent and correct topology. Comparing with other mapping results, NeuroSLAM can also build a 3D experience map of high quality similar to ORB-SLAM. In some cases, it is better than feature-based or direct sparse visual SLAM, such as LDSO.

In addition, we add several quantitative analyses of the geometric accuracy for providing more informative evaluation by computing the ATE and the RE. We extract 19 groups of key positions at the turns in order. Each group consists of five key positions through five trials randomly. The mean of each group of key positions is estimated, as shown in Fig. S5 (Online Appendix). We can see from Fig. S5 (Online Appendix) the majority of the key positions in the NeuroSLAM trajectory are near to the correspondence key positions in the ground truth trajectory. Some key positions in the trajectory generated by LDSO are far from the key correspondence positions in the ground truth trajectory. The main reason is that the map relaxation and odometry measurement in vertical space are not robust in LDSO.

Figure 26 shows the translation error of each group of key positions between the estimated map and the ground truth map, respectively. The translation error of the aidvo becomes larger with the increasing distance traversed due to the accumulate error of the visual odometry. NeuroSLAM has similar accuracy to ORB-SLAM. The accuracy of NeuroSLAM is also higher than LDSO. The quantitative results are consistent with the qualitative results in Fig. 25 and Fig. S5 (Online Appendix).

As shown in Fig. 27, the ATE and RE in overall or subtrajectories between each map with the ground truth map are presented. The NeuroSLAM system has competitive high and stable geometric accuracy as shown in Fig. 27a, b. With increasing distance traversed, the RMSE also increases for all these methods. However, our method has higher geometric accuracy. Compared with the feature-based or direct sparse monocular visual SLAM, our method benefits from both the robust visual odometry and the conjunctive encoding method of experience map. Our visual odometry as introduced in Sect. 4.4 is implemented based on average intensity difference by comparing a set of consecutive images with low resolution, which can work well in extreme conditions, such as quick motion change in a vertical direction. When revisiting a familiar place, the map correction approach can relocate the robot by recalling previous spatial experience. Overall, our method is more robust in variable scenes and extreme conditions.

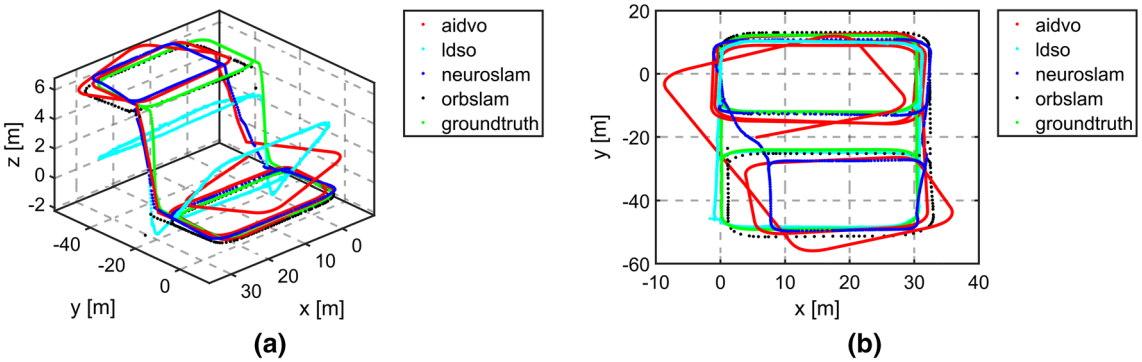

Fig. 25 All 3D mapping results based on several methods. The figures show 3D maps including the aidvo map, the NeuroSLAM map, the ORB-SLAM map, the LDSO map and the ground truth map. **a** 3D view; **b** top view. The figures show the topological consistency with the ground truth map

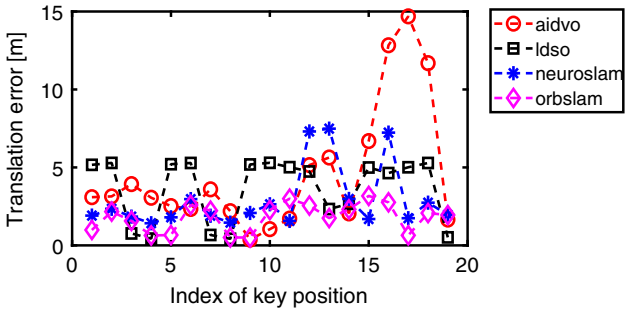

Fig. 26 The translation error of each group of key positions between the NeuroSLAM map, the aidvo map, the LDSO map, the ORB-SLAM map and the ground truth map, respectively

6.3 Demonstration of integration with visual inertial odometry

To demonstrate the 3D mapping performance, we integrate the NeuroSLAM system with a visual inertial odometry. Here, we show some results of NeuroSLAM integrated with a high quality of visual inertial odometry, VINS-mono, which was developed by Qin et al. (2018). We use one of maplab datasets (cla-floor-f) (Schneider et al. 2018) for evaluation. This dataset was collected in a multi-level building. It contains consecutive images captured by monocular camera and IMU sensor data.

As shown in Fig. 28, the topology in the multilayered experience map is consistent with the floor plan in Schneider et al. (2018). The multilayered experience map based on visual inertial odometry has higher geometric accuracy than the vision-only odometry approach as shown in Fig. 28. When revisiting familiar places, the loop closure and experience map relaxation occur. The multilayered experience map becomes stable after moving several loops in 3D environments.

In addition, we add several quantitative analyses of geometric accuracy for providing more informative evaluation by computing the ATE and the RE. We extract eight groups of the key positions at the turns in order. Each group consists of five key positions through five trials randomly. The mean of each group of key positions is estimated, as shown in Fig. S6 (Online Appendix). We can see from Fig. S6 (Online Appendix) the key positions in the NeuroSLAM trajectory are near to the correspondence key positions in the VINS trajectory. As the demonstration of Figs. 29, 30, the Neu-

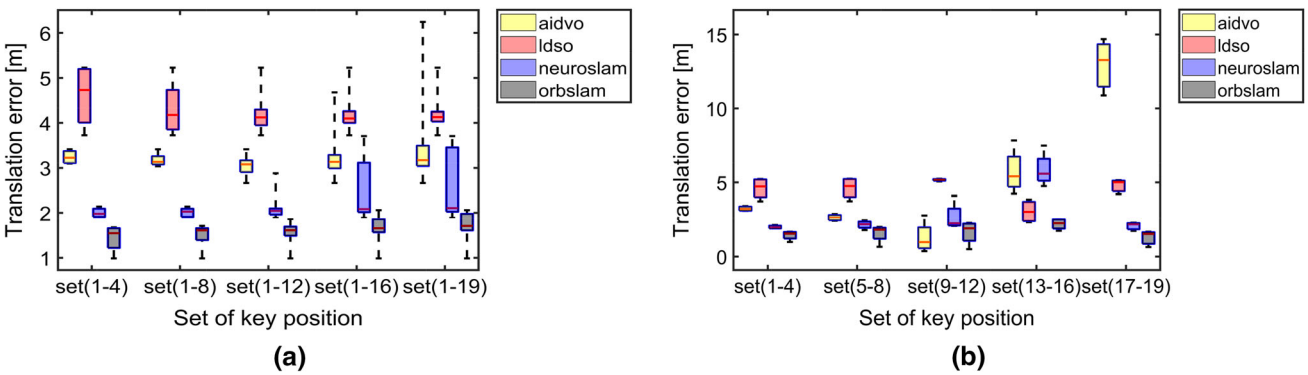

Fig. 27 The ATE and RE of each set of key positions between each map with the ground truth map. **a** The ATE; **b** the RE

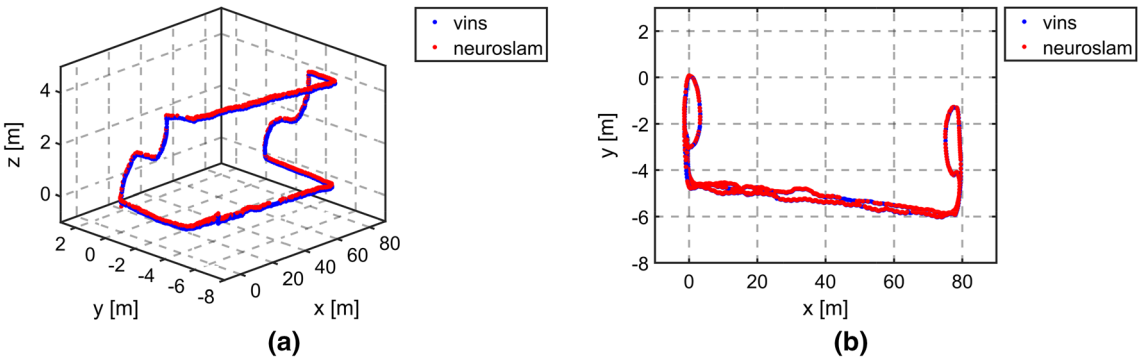

Fig. 28 The multilayered experience map generated from the NeuroSLAM system integrated with VINS and the odometry map based on VINS. **a** 3D view; **b** top view

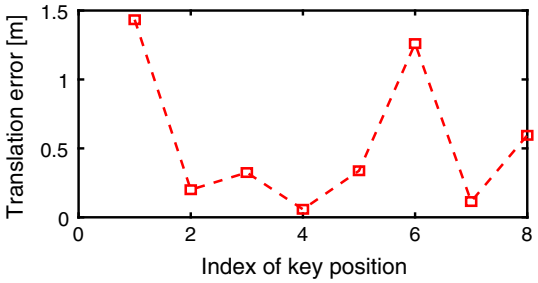

Fig. 29 Translation error of each group of key positions between the NeuroSLAM map and the VINS map

roSLAM system can work well with high geometric accuracy based on the VINS. The translation error is stable and low.

Overall, the NeuroSLAM system is capable of extension, as shown by its integration. Our system can achieve better geometric accuracy and topological consistency when integrated with high-quality odometry. In practice, we need to take many factors into consideration, such as computing load, storage cost and power consumption.

7 Discussion and conclusion

The paper has presented a novel neuro-inspired mapping system with four degrees of freedom for 3D environments known as NeuroSLAM. In the system, we modeled the 3D grid cells and the multilayered head direction cells representing the robot's 4DoF pose in 3D environments. Similar to the mammalian vision perception capability, the system is only coupled with a lightweight vision system that provides external visual cues and self-motion cues. The system built multilayered experience maps with synthetic and real-world datasets consisting by indoor and outdoor parts. The relocalization and loop closure are driven by the multilayered experience map through sequences of familiar visual cues. The experimental results demonstrated the system's capability to generate coherent 3D experience maps with consistent topology in simulated and real-world 3D environments and to process loop closure with significant errors in path integration.

The conjunctive pose cell model combines both 3D grid cells and head direction cells, enabling it to represent a robot's 4DoF pose at an arbitrary 3D location in 3D environments. It is distinctive from the RatSLAM model (Milford et al. 2004; Milford and Wyeth 2008, 2010) which uses pose cells to represent 2D pose (x, y, θ). We modeled the 3D grid cells and multilayered head direction cell, respectively, rather than using a type of combined cell model. Our conjunctive pose cell model can represent a robot's pose when moving through horizontal and vertical space. Comparing with other research (see Sect. 2.2), our model is distinctive and can represent 4DoF pose in 3D space. To the best of our knowledge, the novel discovery of head direction cells and 3D grid cells has not been modeled for 3D SLAM so far. The NeuroSLAM system has some exploratory value in investigating how 5DoF or 6DoF biologically plausible SLAM systems could be implemented.

In mammals, the functional relationship between head direction cells and 3D grid cells is still not entirely clear. In our model, we modeled these two types of cells separately. But a speculative connection between 3D grid cells and head direction cells was also proposed. The system processed the path integration in a 3D grid cell network using the direction information decoded from the multilayered head direction cell network. In order to improve computing efficiency and reduce the complexity of the system, we simplified the neural model. We do not build the 3D place cell model. However, we represent the functional properties of place cells in the 3D grid cell network and the 3D experience map. This model may be helpful to neuroscientists in suggesting experiments for interpreting the neural mechanisms of 3D spatial representation supported by head direction cells and 3D grid cells. As shown in the experimental results of 3D grid cells activity and multilayered head direction cells activity, the trajectory of active cells in the 3D grid cell network has similar char-

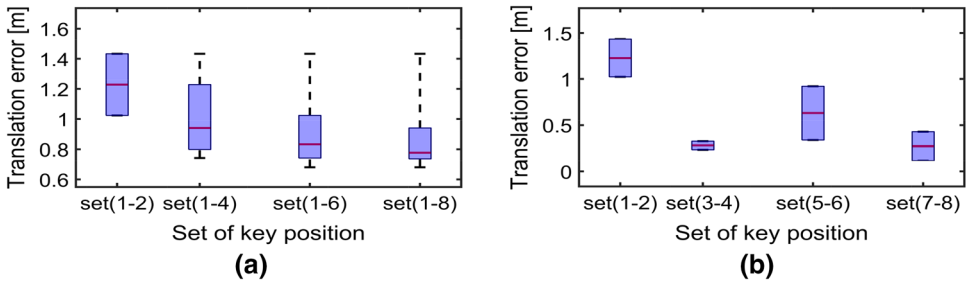

Fig. 30 The ATE and RE of each set of key positions between the NeuroSLAM map and the VINS map. **a** The ATE; **b** the RE

acters of a regular 3D lattice pattern compared with the 3D FCC model (Jeffery et al. 2015; Kim and Maguire 2019). The simplified model of multilayered head direction cells worked well enough for representing 4DoF pose in 3D environment though we did not take incorporate the 3D head direction cells found in mammals (Finkelstein et al. 2015) into consideration. The 3D head direction cells respond to a particular combination of azimuth × pitch, thus representing the direction of the head vector in 3D space (Finkelstein et al. 2015, 2016). Finkelstein et al. (2015) proposed a toroidal model for modeling 3D head direction cells. The model can only represent yaw and pitch. We are looking to expand our model to represent 6DoF pose with complex 3D head direction cells, e.g., a 3D cube head direction network or conjunctive 3D head direction cell network consisting of a toroidal network and a ring network in future work.

NeuroSLAM has some advantages over conventional SLAM methods from the perspective of 3D space encoding, 3D path integration, 3D pose representation and performance. Firstly, we encode the experience map with the conjunctive code of 3D grid cells, head direction cells, and local view cells, which can not only encode a rich spatio-view place experiences, but also maintain some biological plausibility. This encoding method can also reduce false positives and repeatedly correct loop closure even when facing accumulative odometry error. When matching familiar places, we use both the threshold of scene similarity and the distance threshold of conjunctive codes of experiences. In contrast, conventional methods encode places only by geometric coordinates and implement familiar place recognition based on feature matching, which is not as robust in featureless or dynamic environments. Furthermore, NeuroSLAM can reuse existed experiences when revisiting familiar scenes like humans do. However, conventional methods such as ORB-SLAM and LDSO generate a lot of pose nodes continuously along trajectories, which increases the computational complexity and power consumption in large environments. Secondly, the 3D state estimation by path integration (dead reckoning) is a key module in SLAM systems. Conventional SLAM methods are often implemented based on filters or optimization, e.g., ORB-SLAM and LDSO, which assume that the functions of state transition and measurement are linear and the noise is Gaussian. The performance of the SLAM system based on optimization or based on filters, e.g., Kalman filter, extended Kalman filter and particle filter, also suffers when increasing the number of landmarks. All previous landmark estimations are affected with every new added landmark. This can be difficult or even infeasible for long-term task in large complex environments, where the robot faces a huge number of landmarks. Due to these restrictions, these methods may not be capable of performing mapping in real-time, unpredictable environments. However, the NeuroSLAM system could enable robots to locate and map their surrounding environments robustly compared to conventional SLAM methods, since the 3D grid cell model based on the attractor neural network is capable of processing nonlinear state estimation by path integration using neural dynamics in challenging environments, e.g., light or scene changes or quick motion changes. The neural dynamics of excitation and inhibition estimates the robot's pose state reliably by combining self-motion information and local view cues. Thirdly, NeuroSLAM represents a balance between limited 2D RatSLAM-type implementations and full 6DoF implementations like ORB-SLAM. NeuroSLAM is in the middle, and exploits constraints that are reasonable (e.g., no roll) for a range of applications that ORB-SLAM in its current form does not. Finally, the NeuroSLAM model can potentially be deployed using a brain-inspired neuromorphic chip with associated advantages of low power consumption and high computational efficiency in future work. NeuroSLAM's origins in biological inspiration mean that it also has the potential to incorporate further discoveries and mechanisms as they are discovered in the mammalian brain. For instance, we could integrate the NeuroSLAM system with an episodic memory module to improve adaptivity in unpredictable environments. Overall, these properties enable NeuroSLAM to have some competitive advantages over conventional methods. The brain-inspired models show the potential to help further push SLAM to a new level in large, unstructured, unpredictable environments.

Although we use a lightweight visual odometry system here that is only capable of generating relatively coarse estimates of motion with four degrees of freedom, there is the potential to integrate this model with a full 6DoF visual odometry system from the conventional robotics literature such as LIBVISO2 and a multitude of other equivalent tech-

niques. While this may reduce the biological relevance of the results, it will also increase the metric accuracy of the experience maps generated by NeuroSLAM, making it more useful for applications where metric accuracy, especially global metric accuracy, is critical. Likewise, future work could improve loop closure robustness to varying environmental conditions and camera viewpoints by incorporating a more sophisticated visual place recognition process, for example, utilizing semantics and state-of-the-art learned features.

The 3D multilayered experience map generated by the NeuroSLAM system can be learned and generated when the robot visits unknown environments. It can also be maintained and updated based on the learning and recalling mechanism incrementally. The 3D spatial experience nodes represent 4DoF pose in specific 3D location, and the links contain distance and direction between nodes. This metric and topology information can be used for 3D path planning and guidance control in 3D environments. It is likely that map maintenance routines, as implemented in prior work, could also be deployed here to ensure long-term map stability as well as computation and storage viability (Milford and Wyeth 2010). We are looking to test the utility of these experience maps for real robot navigation in future work.

Acknowledgements This work was supported by the National Key Research and Development Program of China (No. 2016YFB0502200), the Fundamental Research Founds for National University, China University of Geo-sciences (Wuhan) (No. 1610491T08) and the Hubei Soft Science Research Program (No. QLZX2014010). MM is also partially supported by an ARC Future Fellowship FT140101229. We thank Sourav Garg and Adam Jacobson for their help in improving the comparison experiments. We appreciate the editor and anonymous reviewers for their insightful comments and suggestions on improving the paper.

References

Arleo A, Gerstner W (2000) Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. Biol Cybern 83(3):287–299. https://doi.org/10.1007/s004220000171

Ball D, Heath S, Wiles J, Wyeth G, Corke P, Milford M (2013) Openratslam: an open source brain-based slam system. Auton Robots 34(3):149–176. https://doi.org/10.1007/s10514-012-9317-9

Banino A, Barry C, Uria B, Blundell C, Lillicrap TP, Mirowski P, Pritzel A, Chadwick MJ, Degris T, Modayil J, Wayne G, Soyer H, Viola F, Zhang B, Goroshin R, Rabinowitz NC, Pascanu R, Beattie C, Petersen S, Sadik A, Gaffney S, King H, Kavukcuoglu K, Hassabis D, Hadsell R, Kumaran D (2018) Vector-based navigation using grid-like representations in artificial agents. Nature 557(7705):429–433. https://doi.org/10.1038/s41586-018-0102-6

Barrera A, Weitzenfeld A (2008) Biologically-inspired robot spatial cognition based on rat neurophysiological studies. Auton Robots 25(1–2):147–169. https://doi.org/10.1007/s10514-007-9074-3

Behley J, Stachniss C (2018) Efficient surfel-based SLAM using 3D laser range data in urban environments. In: Robotics: science and systems. https://doi.org/10.15607/rss.2018.xiv.016

Bellingham J, Dupont PE, Fischer P, Floridi L, Full R, Jacobstein N, Kumar V, McNutt M, Merrifield RD, Nelson BJ, Scassellati B, Taddeo M, Taylor R, Veloso MM, Wang ZL, Wood RJ (2018) The grand challenges of science robotics. Sci Robot 3(14):eaar7650. https://doi.org/10.1126/scirobotics.aar7650

Bjerknes TL, Dagslott NC, Moser EI, Moser MB (2018) Path integration in place cells of developing rats. Proc Natl Acad Sci 115(7):E1637–E1646. https://doi.org/10.1073/pnas.1719054115

Burak Y, Fiete IR (2009) Accurate path integration in continuous attractor network models of grid cells. PLoS Comput Biol 5(2):e1000291. https://doi.org/10.1371/journal.pcbi.1000291

Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I, Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: roward the robust-perception age. IEEE Trans Robot 32(6):1309–1332. https://doi.org/10.1109/tro.2016.2624754

Campbell MG, Ocko SA, Mallory CS, Low IIC, Ganguli S, Giocomo LM (2018) Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation. Nat Neurosci 21(8):1096–1106. https://doi.org/10.1038/s41593-018-0189-y

Casali G, Bush D, Jeffery K (2019) Altered neural odometry in the vertical dimension. In: Proceedings of the national academy of sciences, p 201811867. https://doi.org/10.1073/pnas.1811867116

Cope AJ, Sabo C, Vasilaki E, Barron AB, Marshall JAR (2017) A computational model of the integration of landmarks and motion in the insect central complex. PLOS ONE 12(2):e0172325. https://doi.org/10.1371/journal.pone.0172325

Cummins MJ, Newman P (2008) FAB-MAP: probabilistic localization and mapping in the space of appearance. Int J Robot Res 27(6):647–665. https://doi.org/10.1177/0278364908090961

Davison AJ, Reid ID, Molton ND, Stasse O (2007) MonoSLAM: real-time single camera SLAM. IEEE Trans Pattern Anal Mach Intell 29(6):1052–1067. https://doi.org/10.1109/tpami.2007.1049

Dissanayake MG, Newman P, Clark S, Durrant-Whyte HF, Csorba M (2001) A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans Robot Autom 17(3):229–241. https://doi.org/10.1109/70.938381

Droeschel D, Schwarz M, Behnke S (2017) Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner. Robot Auton Syst 88:104–115. https://doi.org/10.1016/j.robot.2016.10.017

Dupeyroux J, Serres JR, Viollet S (2019) AntBot: a six-legged walking robot able to home like desert ants in outdoor environments. Sci Robot 4(27):eaau0307. https://doi.org/10.1126/scirobotics.aau0307

Endres F, Hess J, Sturm J, Cremers D, Burgard W (2014) 3-D mapping with an RGB-D camera. IEEE Trans Robot 30(1):177–187. https://doi.org/10.1109/tro.2013.2279412

Engel J, Schöps T, Cremers D (2014) LSD-SLAM: large-scale direct monocular SLAM. In: European Conference on computer vision. Springer, Berlin, pp 834–849. https://doi.org/10.1007/978-3-319-10605-2-54

Engel J, Koltun V, Cremers D (2018) Direct sparse odometry. IEEE Trans Pattern Anal Mach Intell 40(3):611–625. https://doi.org/10.1109/tpami.2017.2658577

Evans T, Bicanski A, Bush D, Burgess N (2016) How environment and self-motion combine in neural representations of space. J Physiol 594(22):6535–6546. https://doi.org/10.1113/jp270666

Evers C, Naylor PA (2018) Acoustic SLAM. IEEE/ACM Trans Audio Speech Lang Process 26(9):1484–1498. https://doi.org/10.1109/taslp.2018.2828321

Faessler M, Fontana F, Forster C, Mueggler E, Pizzoli M, Scaramuzza D (2016) Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. J Field Robot 33:431–450. https://doi.org/10.1109/icra.2017.7989679

Finkelstein A, Derdikman D, Rubin A, Foerster JN, Las L, Ulanovsky N (2015) Three-dimensional head-direction coding in the bat brain.

Nature 517(4):159–164. https://doi.org/10.1016/j.cell.2018.09.017
Finkelstein A, Las L, Ulanovsky N (2016) 3-D maps and compasses in the brain. Annu Rev Neurosci 39(1):171–96. https://doi.org/10.1146/annurev-neuro-070815-013831
Finkelstein A, Ulanovsky N, Tsodyks M, Aljadeff J (2018) Optimal dynamic coding by mixed-dimensionality neurons in the head-direction system of bats. Nat Commun 9(1):350. https://doi.org/10.1038/s41467-018-05562-1
Forster C, Pizzoli M, Scaramuzza D (2014) SVO: fast semi-direct monocular visual odometry. In: 2014 IEEE international conference on robotics and automation (ICRA), pp 15–22. https://doi.org/10.1109/icra.2014.6906584
Forster C, Zhang Z, Gassner M, Werlberger M, Scaramuzza D (2017) SVO: semidirect visual odometry for monocular and multicamera systems. IEEE Trans Robot 33(2):249–265. https://doi.org/10.1109/tro.2016.2623335
Gallego G, Lund JEA, Mueggler E, Rebecq H, Delbrück T, Scaramuzza D (2018) Event-based, 6-DOF camera tracking from photometric depth maps. IEEE Trans Pattern Anal Mach Intell 40(10):2402–2412. https://doi.org/10.1109/tpami.2017.2769655
Gao X, Wang R, Demmel N, Cremers D (2018) LDSO: direct sparse odometry with loop closure. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2198–2204
Gaussier P, Banquet JP, Cuperlier N, Quoy M, Aubin L, Jacob PY, Sargolini F, Save E, Krichmar JL, Poucet B (2019) Merging information in the entorhinal cortex: What can we learn from robotics experiments and modeling? J Exp Biol 222(Suppl 1):jeb186932. https://doi.org/10.1242/jeb.186932
Geiger A, Ziegler J, Stiller C (2011) Stereoscan: dense 3D reconstruction in real-time. In: 2011 IEEE intelligent vehicles symposium (IV), pp 963–968. https://doi.org/10.1109/ivs.2011.5940405
Gianelli S, Harland B, Fellous JM (2018) A new rat-compatible robotic framework for spatial navigation behavioral experiments. J Neurosci Methods 294:40–50. https://doi.org/10.1016/j.jneumeth.2017.10.021
Giovannangeli C, Gaussier P (2008) Autonomous vision-based navigation: goal-oriented action planning by transient states prediction, cognitive map building, and sensory-motor learning. In: 2008 IEEE/RSJ International conference on intelligent robots and systems, pp 676–683. https://doi.org/10.1109/iros.2008.4650872
Hafting T, Fyhn M, Molden S, Moser MB, Moser EI (2005) Microstructure of a spatial map in the entorhinal cortex. Nature 436(7052):801–806. https://doi.org/10.1038/nature03721
Hayman RMA, Casali G, Wilson JJ, Jeffery KJ (2015) Grid cells on steeply sloping terrain: evidence for planar rather than volumetric encoding. Front Psychol 6:925. https://doi.org/10.3389/fpsyg.2015.00925
Henry P, Krainin M, Herbst E, Ren X, Fox D (2012) RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. Int J Robot Res 31(5):647–663. https://doi.org/10.1177/0278364911434148
Horiuchi TK, Moss CF (2015) Grid cells in 3-D: reconciling data and models. Hippocampus 25(12):1489–1500. https://doi.org/10.1002/hipo.22469
Jauffret A, Cuperlier N, Gaussier P (2015) From grid cells and visual place cells to multimodal place cell: a new robotic architecture. Front Neurorobot 9:1. https://doi.org/10.3389/fnbot.2015.00001
Jeffery KJ, Jovalekic A, Verriotis M, Hayman R (2013) Navigating in a three-dimensional world. Behav Brain Sci 36(05):523–543. https://doi.org/10.1017/s0140525x12002476
Jeffery KJ, Wilson JJ, Casali G, Hayman RM (2015) Neural encoding of large-scale three-dimensional space-properties and constraints. Front Psychol 6:927. https://doi.org/10.3389/fpsyg.2015.00927
Jeffery KJ, Page HJI, Stringer SM (2016) Optimal cue combination and landmark-stability learning in the head direction system. J Physiol 594(22):6527–6534. https://doi.org/10.1113/jp272945
Karrer M, Schmuck P, Chli M (2018) CVI-SLAM—collaborative visual-inertial SLAM. IEEE Robot Autom Lett 3(4):2762–2769. https://doi.org/10.1109/lra.2018.2837226
Kim M, Maguire EA (2018a) Encoding of 3D head direction information in the human brain. Hippocampus 29:619–629. https://doi.org/10.1002/hipo.23060
Kim M, Maguire EA (2018b) Hippocampus, retrosplenial and parahippocampal cortices encode multicompartment 3D space in a hierarchical manner. Cereb Cortex 28(5):1898–1909. https://doi.org/10.1093/cercor/bhy054
Kim M, Maguire EA (2019) Can we study 3D grid codes non-invasively in the human brain? Methodological considerations and fMRI findings. NeuroImage 186:667–678. https://doi.org/10.1016/j.neuroimage.2018.11.041
Kim M, Jeffery KJ, Maguire EA (2017) Multivoxel pattern analysis reveals 3D place information in the human hippocampus. J Neurosci 37(16):4270–4279. https://doi.org/10.1523/jneurosci.2703-16.2017
Klein G, Murray DW (2007) Parallel tracking and mapping for small AR workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality, pp 225–234. https://doi.org/10.1109/ismar.2007.4538852
Konolige K, Agrawal M (2008) FrameSLAM: from bundle adjustment to real-time visual mapping. IEEE Trans Robot 24(5):1066–1077. https://doi.org/10.1109/tro.2008.2004832
Kreiser R, Cartiglia M, Martel JN, Conradt J, Sandamirskaya Y (2018a) A neuromorphic approach to path integration: a head-direction spiking neural network with vision-driven reset. In: 2018 IEEE international symposium on circuits and systems (ISCAS), pp 1–5. https://doi.org/10.1109/iscas.2018.8351509
Kreiser R, Renner A, Sandamirskaya Y, Pienroj P (2018b) Pose estimation and map formation with spiking neural networks: towards neuromorphic SLAM. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2159–2166. https://doi.org/10.1109/IROS.2018.8594228
Krombach N, Droeschel D, Houben S, Behnke S (2018) Feature-based visual odometry prior for real-time semi-dense stereo SLAM. Robot Auton Syst 109:38–58. https://doi.org/10.1016/j.robot.2018.08.002
Kropff E, Carmichael JE, Moser MB, Moser EI (2015) Speed cells in the medial entorhinal cortex. Nature 523(7561):419–424. https://doi.org/10.1038/nature14622
Laurens J, Angelaki DE (2018) The brain compass: a perspective on how self-motion updates the head direction cell attractor. Neuron 97(2):275–289. https://doi.org/10.1016/j.neuron.2017.12.020
Laurens J, Kim B, Dickman JD, Angelaki DE (2016) Gravity orientation tuning in macaque anterior thalamus. Nat Neurosci 19(12):1566–1568. https://doi.org/10.1038/nn.4423
Lever C, Burton S, Jeewajee A, O'Keefe J, Burgess N (2009) Boundary vector cells in the subiculum of the hippocampal formation. J Neurosci 29(31):9771–9777. https://doi.org/10.1523/jneurosci.1319-09.2009
Llofriu M, Tejera G, Contreras M, Pelc T, Fellous J, Weitzenfeld A (2015) Goal-oriented robot navigation learning using a multi-scale space representation. Neural Netw 72:62–74. https://doi.org/10.1016/j.neunet.2015.09.006
Lowry SM, Sünderhauf N, Newman P, Leonard JJ, Cox DD, Corke PI, Milford M (2016) Visual place recognition: a survey. IEEE Trans Robot 32(1):1–19. https://doi.org/10.1109/tro.2015.2496823
Lynen S, Bosse M, Siegwart R (2016) Keyframe-based visual–inertial odometry using nonlinear optimization. Int J Robot Res 124(1):49–64. https://doi.org/10.1007/s11263-016-0947-9

Maddern WP, Milford M, Wyeth G (2012) CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory. Int J Robot Res 31(4):429–451. https://doi.org/10.1177/0278364912438273

Matsuki H, von Stumberg L, Usenko VC, Stuckler J, Cremers D (2018) Omnidirectional DSO: direct sparse odometry with fisheye cameras. IEEE Robot Autom Lett 3(4):3693–3700. https://doi.org/10.1109/lra.2018.2855443

McNaughton BL, Battaglia FP, Jensen O, Moser EI, Moser MB (2006) Path integration and the neural basis of the 'cognitive map'. Nat Rev Neurosci 7(8):663–678. https://doi.org/10.1038/nrn1932

Meyer JA, Guillot A, Girard B, Khamassi M, Pirim P, Berthoz A (2005) The Psikharpax project: towards building an artificial rat. Robot Auton Syst 50(4):211–223. https://doi.org/10.1016/j.robot.2004.09.018

Milford M (2013) Vision-based place recognition: How low can you go? Int J Robot Res 32(7):766–789. https://doi.org/10.1177/0278364913490323

Milford M, Schulz R (2014) Principles of goal-directed spatial robot navigation in biomimetic models. Philos Trans R Soc B Biol Sci 369(1655):20130484. https://doi.org/10.1098/rstb.2013.0484

Milford M, Wyeth G (2008) Mapping a suburb with a single camera using a biologically inspired SLAM system. IEEE Trans Robot 24(5):1038–1053. https://doi.org/10.1109/tro.2008.2004520

Milford M, Wyeth G (2010) Persistent navigation and mapping using a biologically inspired SLAM system. Int J Robot Res 29(9):1131–1153. https://doi.org/10.1016/j.robot.2010.05.004

Milford M, Wyeth G (2012) SeqSLAM: visual route-based navigation for sunny summer days and stormy winter nights. In: 2012 IEEE international conference on robotics and automation, pp 1643–1649. https://doi.org/10.1109/icra.2012.6224623

Milford MJ, Wyeth GF, Prasser D (2004) RatSLAM: a hippocampal model for simultaneous localization and mapping. In: 2004 IEEE international conference on robotics and automation (ICRA). IEEE, vol 1, pp 403–408. https://doi.org/10.1109/robot.2004.1307183

Milford M, McKinnon D, Warren M, Wyeth G, Upcroft B (2011a) Feature-based visual odometry and featureless place recognition for SLAM in 2.5 d environments. In: In Drummond, Tom (eds.) ACRA 2011 Proceedings, Australian robotics & automation association, robotics: science and systems foundation, pp 1–8. https://doi.org/10.15607/rss.2013.ix.003

Milford M, Schill F, Corke PI, Mahony RE, Wyeth G (2011b) Aerial SLAM with a single camera using visual expectation. In: 2011 IEEE international conference on robotics and automation, pp 2506–2512. https://doi.org/10.1109/icra.2011.5980329

Montemerlo M, Thrun S, Koller D, Wegbreit B, et al (2002) FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of the national conference on artificial intelligence (AAAI)

Moser EI, Moser MB, McNaughton BL (2017) Spatial representation in the hippocampal formation: a history. Nat Neurosci 20(11):1448–1464. https://doi.org/10.1038/nn.4653

Mulas M, Waniek N, Conradt J (2016) Hebbian plasticity realigns grid cell activity with external sensory cues in continuous attractor models. Front Comput Neurosci 10:13. https://doi.org/10.3389/fncom.2016.00013

Mur-Artal R, Tardós JD (2017) Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans Robot 33(5):1255–1262. https://doi.org/10.1109/tro.2017.2705103

Mur-Artal R, Montiel JMM, Tardos JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot 31(5):1147–1163. https://doi.org/10.1109/tro.2015.2463671

Naseer T, Burgard W, Stachniss C (2018) Robust visual localization across seasons. IEEE Trans Robot 34(2):289–302. https://doi.org/10.1109/tro.2017.2788045

Newcombe RA, Lovegrove S, Davison AJ (2011) DTAM: dense tracking and mapping in real-time. In: 2011 International conference on computer vision, pp 2320–2327. https://doi.org/10.1109/iccv.2011.6126513

O'Keefe J, Dostrovsky J (1971) The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. Brain Res 34(1):171–175. https://doi.org/10.1016/0006-8993(71)90358-1

Page HJI, Wilson JJ, Jeffery KJ (2018) A dual-axis rotation rule for updating the head direction cell reference frame during movement in three dimensions. J Neurophysiol 119(1):192–208. https://doi.org/10.1152/jn.00501.2017

Paul R, Newman P (2010) FAB-MAP 3D: topological mapping with spatial and visual appearance. In: 2010 IEEE international conference on robotics and automation, pp 2649–2656. https://doi.org/10.1109/robot.2010.5509587

Qin T, Li P, Shen S (2018) Vins-mono: a robust and versatile monocular visual–inertial state estimator. IEEE Trans Robot 34(4):1004–1020. https://doi.org/10.1109/tro.2018.2853729

Rebecq H, Horstschaefer T, Gallego G, Scaramuzza D (2017) EVO: a geometric approach to event-based 6-DOF parallel tracking and mapping in real time. IEEE Robot Autom Lett 2(2):593–600. https://doi.org/10.1109/lra.2016.2645143

Sabo CM, Cope A, Gurney K, Vasilaki E, Marshall J (2016) Bio-inspired visual navigation for a quadcopter using optic flow. In: AIAA Infotech @ Aerospace, American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2016-0404

Sabo C, Yavuz E, Cope A, Gumey K, Vasilaki E, Nowotny T, Marshall JAR (2017) An inexpensive flying robot design for embodied robotics research. In: 2017 International joint conference on neural networks (IJCNN), IEEE. IEEE, pp 4171–4178. https://doi.org/10.1109/ijcnn.2017.7966383

Samsonovich A, McNaughton BL (1997) Path integration and cognitive mapping in a continuous attractor neural network model. J Neurosci 17(15):5900–5920. https://doi.org/10.1523/jneurosci.17-15-05900.1997

Saputra MRU, Markham A, Trigoni N (2018) Visual SLAM and structure from motion in dynamic environments: a survey. ACM Comput Surv 51(2):1–36. https://doi.org/10.1145/3177853

Schneider T, Dymczyk M, Fehr M, Egger K, Lynen S, Gilitschenski I, Siegwart R (2018) maplab: an open framework for research in visual–inertial mapping and localization. IEEE Robot Autom Lett 3(3):1418–1425

Shinder ME, Taube JS (2019) Three-dimensional tuning of head direction cells in rats. J Neurophysiol 121(1):4–37. https://doi.org/10.1152/jn.00880.2017

Shipston-Sharman O, Solanka L, Nolan MF (2016) Continuous attractor network models of grid cell firing based on excitatory–inhibitory interactions. J Physiol 594(22):6547–6557. https://doi.org/10.1113/jp270630

Silveira L, Guth F, Drews P, Botelho S (2013) 3D robotic mapping: a biologic approach. In: 2013 16th international conference on advanced robotics (ICAR), IEEE. IEEE, pp 1–6. https://doi.org/10.1109/icar.2013.6766531

Silveira L, Guth F, Drews-Jr P, Ballester P, Machado M, Codevilla F, Duarte-Filho N, Botelho S (2015) An open-source bio-inspired solution to underwater SLAM. IFAC-PapersOnLine 48(2):212–217. https://doi.org/10.1016/j.ifacol.2015.06.035

Solstad T, Boccara CN, Kropff E, Moser MB, Moser EI (2008) Representation of geometric borders in the entorhinal cortex. Science 322(5909):1865–1868. https://doi.org/10.1126/science.1166466

Soman K, Chakravarthy S, Yartsev MM (2018) A hierarchical anti-Hebbian network model for the formation of spatial cells in three-dimensional space. Nat Commun 9(1):4046. https://doi.org/10.1038/s41467-018-06441-5

Stackman RW, Tullman ML, Taube JS (2000) Maintenance of rat head direction cell firing during locomotion in the vertical plane. J Neurophysiol 83(1):393–405. https://doi.org/10.1152/jn.2000.83.1.393

Steckel J, Peremans H (2013) BatSLAM: simultaneous localization and mapping using biomimetic sonar. PLoS ONE 8(1):e54076. https://doi.org/10.1371/journal.pone.0054076

Stone T, Differt D, Milford M, Webb B (2016) Skyline-based localisation for aggressively manoeuvring robots using UV sensors and spherical harmonics. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 5615–5622. https://doi.org/10.1109/icra.2016.7487780

Tang G, Michmizos KP (2018) Gridbot: an autonomous robot controlled by a spiking neural network mimicking the brain's navigational system. In: Proceedings of the international conference on neuromorphic systems, ACM. ACM Press. https://doi.org/10.1145/3229884.3229888

Tang H, Yan R, Tan KC (2018) Cognitive navigation by neuro-inspired localization, mapping, and episodic memory. IEEE Trans Cogn Dev Syst 10(3):751–761. https://doi.org/10.1109/tcds.2017.2776965

Taube J, Muller R, Ranck J (1990) Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. J Neurosci 10(2):420–435. https://doi.org/10.1523/jneurosci.10-02-00420.1990

Thrun S, Leonard JJ (2008) Simultaneous localization and mapping. In: Springer Handbook of Robotics, Springer, Berlin, pp 871–889. https://doi.org/10.1007/978-3-540-30301-5-38

Thrun S, Montemerlo M (2006) The graph SLAM algorithm with applications to large-scale mapping of urban structures. Int J Robot Res 25(5–6):403–429. https://doi.org/10.1177/0278364906065387

Vidal AR, Rebecq H, Horstschaefer T, Scaramuzza D (2018) Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. IEEE Robot Autom Lett 3(2):994–1001. https://doi.org/10.1109/lra.2018.2793357

Welchman AE (2016) The human brain in depth: How we see in 3D. Annu Rev Vis Sci 2(1):345–376. https://doi.org/10.1146/annurev-vision-111815-114605

Wohlgemuth MJ, Yu C, Moss CF (2018) 3D hippocampal place field dynamics in free-flying echolocating bats. Front Cell Neurosci 12:270. https://doi.org/10.3389/fncel.2018.00270

Yartsev MM, Ulanovsky N (2013) Representation of three-dimensional space in the hippocampus of flying bats. Science 340(6130):367–372. https://doi.org/10.1126/science.1235338

Zeng T, Si B (2017) Cognitive mapping based on conjunctive representations of space and movement. Front Neurorobot 11:61. https://doi.org/10.3389/fnbot.2017.00061

Zhang Z, Scaramuzza D (2018) A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 7244–7251. https://doi.org/10.1109/IROS.2018.8593941

Zhang Z, Rebecq H, Forster C, Scaramuzza D (2016) Benefit of large field-of-view cameras for visual odometry. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 801–808. https://doi.org/10.1109/icra.2016.7487210

Zhou X, Weber C, Wermter S (2018) A self-organizing method for robot navigation based on learned place and head-direction cells. In: 2018 International joint conference on neural networks (IJCNN). IEEE, pp 1–8. https://doi.org/10.1109/ijcnn.2018.8489348