Autonomous Robots (2020) 44:927–946
https://doi.org/10.1007/s10514-020-09905-0

Improved and scalable online learning of spatial concepts and language models with mapping

Akira Taniguchi[1] · Yoshinobu Hagiwara[1] · Tadahiro Taniguchi[1] · Tetsunari Inamura[2]

Received: 27 December 2018 / Accepted: 24 January 2020 / Published online: 8 February 2020

Abstract
We propose a novel online learning algorithm, called SpCoSLAM 2.0, for spatial concepts and lexical acquisition with high accuracy and scalability. Previously, we proposed SpCoSLAM as an online learning algorithm based on unsupervised Bayesian probabilistic model that integrates multimodal place categorization, lexical acquisition, and SLAM. However, our original algorithm had limited estimation accuracy owing to the influence of the early stages of learning, and increased computational complexity with added training data. Therefore, we introduce techniques such as fixed-lag rejuvenation to reduce the calculation time while maintaining an accuracy higher than that of the original algorithm. The results show that, in terms of estimation accuracy, the proposed algorithm exceeds the original algorithm and is comparable to batch learning. In addition, the calculation time of the proposed algorithm does not depend on the amount of training data and becomes constant for each step of the scalable algorithm. Our approach will contribute to the realization of long-term spatial language interactions between humans and robots.

Keywords Online learning · Place categorization · Scalability · Semantic mapping · Lexical acquisition · Unsupervised Bayesian probabilistic model

1 Introduction

Robots operating in various human environments must adaptively and sequentially acquire new categories for places and unknown words related to various places as well as the map of the environment (Kostavelis and Gasteratos 2015). It is desirable for robots to acquire place categories and vocabulary autonomously based on their experience because it is difficult to manually design spatial knowledge in advance. Related research in the fields of semantic mapping and place categorization (Pronobis and Jensfelt 2012; Kostavelis and Gasteratos 2015; Sünderhauf et al. 2016; Landsiedel et al. 2017; Rangel et al. 2018) has attracted considerable interest in recent years. However, conventional approaches in most of these studies are limited insofar as the robots cannot learn unknown words and unknown place categories without preset vocabulary and categories. In addition, the processes for Simultaneous Localization And Mapping (SLAM) (Thrun et al. 2005) and for estimating semantics related to place have been addressed as separated module processes. However, in our proposed approach, the robot can automatically and simultaneously perform place categorization and environment mapping, and it can learn unknown words without prior knowledge. Our previously proposed unsupervised Bayesian probabilistic model integrates multimodal place categorization, lexical acquisition, and SLAM. In particular, this paper focuses on the problems of estimation accuracy and computational scalability in online learning.

This work was partially supported by JST CREST Grant Number JPMJCR15E3, and JSPS KAKENHI Grant Numbers JP17J07842, JP16H06561, and JP16K12497.

✉ Akira Taniguchi
a.taniguchi@em.ci.ritsumei.ac.jp

Yoshinobu Hagiwara
yhagiwara@em.ci.ritsumei.ac.jp

Tadahiro Taniguchi
taniguchi@em.ci.ritsumei.ac.jp

Tetsunari Inamura
inamura@nii.ac.jp

[1] Ritsumeikan University, 1-1-1 Noji-Higashi, Kusatsu, Shiga 525-8577, Japan

[2] The National Institute of Informatics / SOKENDAI (The Graduate University for Advanced Studies), 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

We define a spatial concept as a place category is autonomously learned by the robot based on multimodal perceptual information, which includes names of places, features of scene images, and position distributions. Then, we define a position distribution as the spatial extent representing a place in the environment. Our study regarding the spatial concept formation and the lexical acquisition also constitute constructive approaches to the human developmental process and symbol emergence in cognitive developmental systems (Cangelosi and Schlesinger 2015; Taniguchi et al. 2018b). Thus, we assume that the robot has not acquired any vocabulary in advance and can recognize only phonemes or syllables. In addition, the robot does not have prior knowledge of the current environment. In this study, a scenario in which the user teaches the robot the name of a place using a spoken utterance while moving together in the environment is studied. An overview of the scenario for online learning task is shown in Fig. 1. The robot and the user move around the environment. When they come to a place where the user wishes to teach, the user speaks a sentence regarding the place to the robot. The robot recognizes the speech, including unknown words, and segments the speech into words. Then, the robot obtains the present estimated position, the scene image, and the speech signal at that time, and acquires spatial knowledge regarding the environment, such as the relationship between words and places.

In online learning, also called sequential learning or incremental learning, an increase in scalability without reducing accuracy is especially important but difficult to achieve for mobile robots. Online learning has the advantage of being performed in real-time. This means that it can be used to adapt immediately to new data by sequentially estimating parameters each time. On the other hand, batch learning takes time to collect large amounts of data and to iterate it for learning. In the case of online learning, previous knowledge can be used immediately for reasoning and tasks such as language communication. Taniguchi et al. (2017) focused on deriving and constructing an appropriate online learning algorithm mathematically based on a theory of machine learning. In our previous work, we proposed SpCoSLAM as an integrated model of nonparametric Bayesian multimodal categorization, a Bayesian filter-based SLAM, speech recognition, and word segmentation, from the standpoint of unsupervised machine learning. However, this algorithm (Taniguchi et al. 2017) had inferior accuracy in terms of categorization and word segmentation compared to batch learning, owing to a situation whereby sufficient statistical information could not be used at the early stages of learning. In addition, speech recognition and unsupervised word segmentation were not completely online, and batch learning was used as an approximation. Therefore, the computational complexity of the processes of speech recognition and unsupervised word segmentation increased with an increase in training data. To enable online learning based on long-term human–robot interactions with limited computational resources, the following core problems need to be solved: (i) the increase in calculation cost owing to an increase in data, and (ii) the decrease in estimation accuracy when compared with batch learning. In intelligent robotics, the framework of online learning is regarded as important. In particular, online learning, which solves the above problem, is required for robots that gain knowledge while moving in the real world.

We here describe improved and scalable algorithms to solve the above-mentioned problems. The improved algorithm mainly addresses the problems of misrecognition (misclassification) and word segmentation in online learning. The scalable algorithm mainly addresses the problem of the increase in computation time. In this study, we introduce the approach of fixed-lag rejuvenation, which is considered particularly effective at solving these problems. Regarding the problem of online lexical acquisition, the improved and scalable algorithms take two respective approaches to the solution. The improved algorithm addresses the problem of under-segmentation, whereby the phoneme sequence is insufficiently segmented, by changing the manner by which the language model is updated such that it re-segments the word sequence. The scalable algorithm performs in a pseudo-online manner by introducing a fixed-lag rejuvenation approach to speech recognition and word segmentation.

One of the advantages to the proposed online learning algorithm is that spatial concepts mistakenly learned by the robot can be corrected sequentially, something that could not be achieved thus far. Moreover, with the proposed algorithm, the robot can flexibly deal with changes in the environment and the names of places. The lower part in Fig. 1 shows the progress of online learning. In the lower left of Fig. 1, clustered places and words are incorrectly estimated, as shown by the elongated purple and blue ellipses. In the lower right of Fig. 1, by contrast, more accurate estimation is achieved by correcting errors as learning progresses. This is realized by reviewing and rethinking previous estimation results when new data is obtained.

The main contributions of this paper are as follows:

- We propose an improved and scalable online learning algorithm with several novel techniques such as fixed-lag rejuvenation.
- The improved online algorithm achieves an accuracy of place categorization and lexical acquisition comparable to batch learning.
- The scalable online algorithm achieves faster learning compared to original algorithms by reducing the order of computational complexity.

The remainder of this paper is organized as follows. In Sect. 2, we discuss related work on the formation of spatial

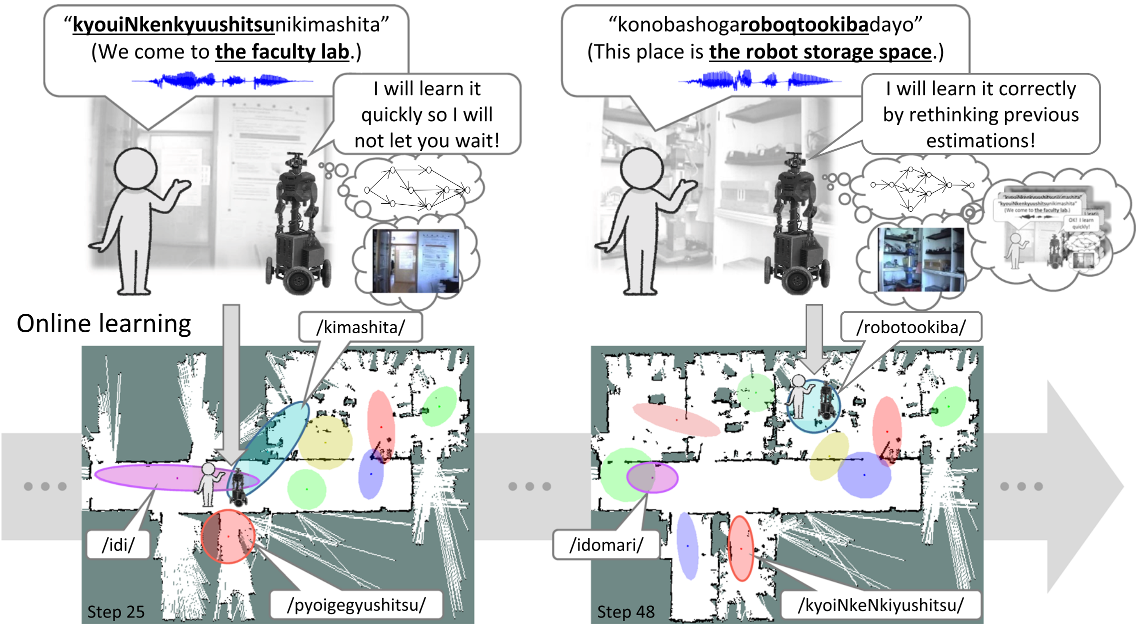

"kyouiNkenkyuushitsunikimashita"
(We come to the faculty lab.)
I will learn it quickly so I will not let you wait!
"konobashogaroboqtookibadayo"
(This place is the robot storage space.)
I will learn it correctly by rethinking previous estimations!
Online learning
/kimashita/
/idi/
/pyoigegyushitsu/
Step 25
/robotookiba/
/idomari/
/kyoiNkeNkiyushitsu/
Step 48

Fig. 1 Overview of the scenario for online learning in this study. We assume a scenario in which the user teaches the robot the name of the place using a spoken utterance while moving together in the environment. The robot learns spatial concepts, language models, and maps while sequentially correcting mistakes from previous learnings based on its interaction with the user and environment, as shown from the bottom left to the bottom right

concepts and online learning that is relevant to our study. In Sect. 3, we present an overview of the model, along with the formulation and the original online learning algorithm, SpCoSLAM. In Sect. 4, we present our proposed algorithms for improved and scalable online learning. In Sect. 5, we discuss the effectiveness of the proposed algorithms in a real environment. In Sect. 6, we evaluate the performance of place categorization and lexical acquisition in various virtual home environments. Section 7 concludes the paper.

2 Related work

2.1 Spatial concept formation

Taguchi et al. (2011) proposed an unsupervised method for simultaneously categorizing self-positions and phoneme sequences from user speech without any prior language model. Taniguchi et al. (2016, 2018a) proposed the nonparametric Bayesian Spatial Concept Acquisition method (SpCoA) using an unsupervised word segmentation method, latticelm (Neubig et al. 2012), and SpCoA++ for highly accurate lexical acquisition as a result of updating the language model. Gu et al. (2016) proposed a method to learn relative spatial concepts, i.e., the words related to distance and direction, from the positional relationship between an utterer and objects. Isobe et al. (2017) proposed a learning method to derive the relationship between objects and places using image features obtained by a Convolutional Neural Network (CNN) (Krizhevsky et al. 2012). Hagiwara et al. (2018) implemented a hierarchical clustering method for the formation of hierarchical place concepts. However, none of the above methods can sequentially learn spatial concepts from unknown environments without a map, because they rely on batch-learning algorithms. Therefore, we developed in previous work an online algorithm, SpCoSLAM (Taniguchi et al. 2017), that can sequentially learn a map, a lexicon, and spatial concepts to integrate positions, speech signals, and scene images. In Taniguchi et al. (2017), however, the accuracy was inferior to that of SpCoA. In this paper, we also compare our proposal to the latest batch learning method, SpCoA++. Because SpCoA++ is able to achieve nearly correct lexical acquisition, if we can successfully overcome the above problems by appropriately devising the learning algorithm, its accuracy should improve even with online lexical acquisition.

Our approach is relevant to research integrating semantic mapping with natural language processing (Walter et al. 2013; Hemachandra et al. 2014). Walter et al. (2013) developed an algorithm that can learn semantic graphs to integrate semantic representation into metric maps from natural language descriptions of aspects such as labels and spatial relationships. Hemachandra et al. (2014) proposed a mechanism to more effectively ground natural language descriptions by integrating scene appearance observations using camera images and laser data. In these studies, a word list, place labels, and the number of category types were known in advance. However, it is challenging to sequentially acquire new words and categories efficiently from a situation in which the lists of words and categories are not provided in advance. Our study includes lexical acquisition for unknown words and formation of new categories from speech signals using spatial information.

Ball et al. (2013) implemented a biologically inspired mapping system, RatSLAM, which is related to pose cells in the hippocampus of a rodent. In addition, robots called Lingodroids using RatSLAM could acquire a lexicon related to places through robot-to-robot communication (Heath et al. 2016). These studies reported that robots created their own vocabulary. Ueda et al. (2016) proposed a brain-inspired method, namely, a Particle Filter on Episode (PFoE) for agent decision making. PFoE can estimate the agent's internal state based on previous events recalled at the time. All previous data is thus accumulated to construct a state space in PFoE. We believe that PFoE is unsuitable for long-term trials because the state space becomes enormous. By contrast, our approach forms concepts from episodes using resources more reasonably for calculations, insofar as the state space is reduced through clustering. Although our proposed method was not originally inspired by biology or brain science, such research is highly suggestive. SpCoSLAM is an integrated model of self-localization, mapping, concept formation, and lexical acquisition. From the point of view of the brain, it may be possible to regard SpCoSLAM as a model that imitates some functions of the hippocampus and the cerebral cortex. If we assume that the training data—i.e., the robot's experiences based on a user's utterances—is the episodic memory, and that spatial concepts are semantic memory, the proposed algorithm can be interpreted as a representation of the process of forming concepts by extracting meaning from short-term episodic memory sequentially. Such matters are not further discussed in this paper, although they remain important for future research.

2.2 Improvement of online learning based on particle filters in unsupervised Bayesian models

As an approach involving Bayesian models that is similar to our model, there are related studies on object concepts. In particular, Araki et al. (2012b) proposed online Multimodal Latent Dirichlet Allocation (oMLDA) to acquire object concepts in an online manner, and combined this with the Nested Pitman–Yor Language Model (NPYLM), making it possible to perform lexical acquisition of unknown words sequentially. Aoki et al. (2016) constructed an algorithm that can infer an approximately global optimal solution by representing it as a single integrated model. The NPYLM is an unsupervised morphological analysis method based on a statistical model that enables word segmentation exclusively from phoneme sequences (Mochihashi et al. 2009). In addition, Nishihara et al. (2017) was able to reduce phoneme recognition errors by applying PFoMDLA to inferences using a particle filter instead of oMLDA. In these studies, online learning was realized as an algorithm in unsupervised machine learning. A spatial concept requires more real-time processing than an object concept because the robot learns spatial concepts while it moves through the environment. The mobile robot should not halt its spatial movement for calculations. Therefore, a more efficient and scalable algorithm is required.

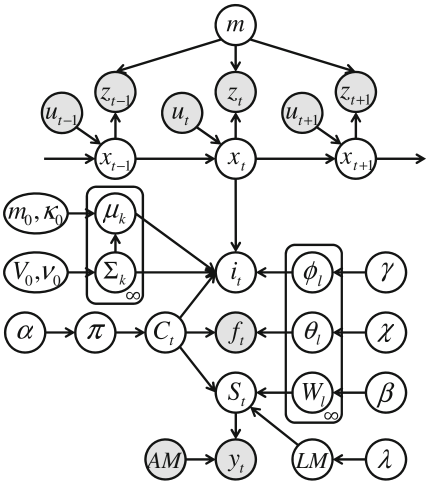
Canini et al. (2009) improved the accuracy of an online algorithm based on a particle filter with the rejuvenation technique. This technique resamples some randomly selected samples of previous observation data from a conditional probabilistic distribution similar to Gibbs sampling. For a completely random choice, the robot needs to memorize all of the previous data. Rejuvenation can deal with the problem of degenerating particles in particle filters. In this study, we introduce rejuvenation into our SpCoSLAM online learning algorithm. In our algorithm, we perform resampling from some recent data. Therefore, we consider that it will be possible to improve the estimation accuracy efficiently.

As another particle filter approach, Börschinger and Johnson (2011) proposed an online algorithm based on a Bayesian model for word segmentation. In addition, Börschinger and Johnson (2012) presented an incremental learning algorithm that introduces rejuvenation to a particle filter. They improved the performance of word segmentation with higher accuracy. The studies above were premised on segmentation of sequences without phoneme recognition errors. In this study, by contrast, the online word segmentation task is particularly challenging because phoneme recognition errors are included in speech recognition results.

3 SpCoSLAM: Online learning for spatial concepts and lexical acquisition with mapping

3.1 Overview

SpCoSLAM has the advantage that spatial concept formation, lexical acquisition, and SLAM, can be performed simultaneously by an integrated model. Figure 2 shows the graphical model of SpCoSLAM and lists each variable of the graphical model. The details of the formulation of the generation process represented by the graphical model are described in Taniguchi et al. (2017). The method learns sequential spatial concepts for unknown environments without maps. It also learns the many-to-many correspondences between places and words via spatial concepts and can mutually complement



Symbol	Definition
m	Environmental map
x_t	Self-position of a robot
z_t	Depth data
u_t	Control data
f_t	Image feature
y_t	Speech signal
i_t	Index of position distributions
C_t	Index of spatial concepts
S_t	Word sequence (word segmentation result)
μ_k, Σ_k	Parameters of Gaussian distribution (position distribution)
π	Parameter of multinomial distribution for index C_t of spatial concepts
ϕ_l	Parameter of multinomial distribution for index i_t of the position distribution
θ_l	Parameter of multinomial distribution for the image feature
W_l	Parameter of multinomial distribution for the names of places
LM	Language model (word dictionary)
AM	Acoustic model for speech recognition
α, β, γ, χ, λ, m_0, κ_0, V_0, ν_0	Hyperparameters of prior distributions

Fig. 2 Left: graphical model representation of SpCoSLAM (Taniguchi et al. 2017). Gray nodes indicate observation variables, and white nodes are unobserved latent variables. Right: description of random variables in SpCoSLAM

the uncertainty of information using multimodal information. Furthermore, the proposed method estimates an appropriate number of clusters of spatial concepts and position distributions depending on the data by using the so-called online Chinese Restaurant Process (CRP) (Aldous 1985), one of the constitutive methods of the Dirichlet Process (DP). In addition, lexical acquisition including unknown words is possible by sequentially updating the language model.

The procedure of SpCoSLAM for each step is described as follows. (a) The robot obtains Weighted Finite-State Transducer (WFST) speech recognition results from the user's speech signals using a language model. (b) The robot obtains the likelihood of self-localization by performing FastSLAM. (c) The robot segments the WFST speech recognition results using an unsupervised word segmentation approach called latticelm (Neubig et al. 2012). (d) The robot obtains the latent variables of spatial concepts by sampling. (e) The robot obtains the marginal likelihood of the observed data as the importance weight. (f) The robot updates the environmental map. (g) The robot estimates the set of model parameters of the spatial concepts from the observed data and the sampled variables. (h) The robot updates the language model of the maximum weight for the next step. (i) The particles are resampled according to their weights. Steps (b)–(g) are performed for each particle.

3.2 Formulation of the online learning algorithm

Our previously proposed online learning algorithm, SpCoSLAM, introduces sequential equation updates to estimate the parameters of the spatial concepts into the formulation of a Rao-Blackwellized Particle Filter (RBPF) (Doucet et al. 2000) in the FastSLAM 2.0 algorithm, which is landmark-based SLAM (Montemerlo et al. 2003), and the technique (Grisetti et al. 2007) applied to grid-based SLAM in a similar manner to that in FastSLAM 2.0. The particle filter is advantageous in that parallel processing can be easily applied because each particle can be calculated independently.

In the formulation of SpCoSLAM, the joint posterior distribution can be factorized to the probability distributions of a language model LM, a map m, the set of model parameters of spatial concepts $\Theta = \{\mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \theta, \phi, \pi\}$, the joint distribution of the self-position trajectory $x_{0:t}$, and the set of latent variables $\mathbf{C}_{1:t} = \{i_{1:t}, C_{1:t}, S_{1:t}\}$. We describe the joint posterior distribution of SpCoSLAM as follows:

$$\begin{aligned} & p(x_{0:t}, \mathbf{C}_{1:t}, LM, \Theta, m \mid u_{1:t}, z_{1:t}, y_{1:t}, f_{1:t}, AM, \mathbf{h}) \\ & \quad = p(LM \mid S_{1:t}, \lambda) p(\Theta \mid x_{0:t}, \mathbf{C}_{1:t}, f_{1:t}, \mathbf{h}) p(m \mid x_{0:t}, z_{1:t}) \\ & \quad \cdot \underbrace{p(x_{0:t}, \mathbf{C}_{1:t} \mid u_{1:t}, z_{1:t}, y_{1:t}, f_{1:t}, AM, \mathbf{h})}_{\text{Particle filter}} \qquad (1) \end{aligned}$$

where the set of hyperparameters is denoted by $\mathbf{h} = \{\alpha, \beta, \gamma, \chi, \lambda, m_0, \kappa_0, V_0, \nu_0\}$. It is noteworthy that the speech signal y_t is not observed during all time-steps. Herein, the proposed method is equivalent to FastSLAM 2.0 when y_t is not observed, i.e., when the speech signal is a trigger for the place categorization.

3.2.1 Particle filter algorithm

The particle filter algorithm uses Sampling Importance Resampling (SIR). The importance weight is denoted by

$\omega_t^{[r]} = P_t^{[r]}/Q_t^{[r]}$ for each particle, where r is the particle index. The target distribution is $P_t^{[r]}$, and the proposal distribution is $Q_t^{[r]}$. The number of particles is R. The following equations are also calculated for each particle r; however, the subscripts representing the particle index are omitted.

We apply two modifications related to the weighting of the original SpCoSLAM algorithm (Taniguchi et al. 2017): (i) additional weight for i_t, C_t, and x_t (AW), and (ii) weight for selecting a language model LM (WS). These modifications are more theoretically reasonable than the original SpCoSLAM model, and our proposed SpCoSLAM 2.0 online learning algorithm is extended on their basis.

We describe the target distribution P_t that modified the derivation of Taniguchi et al. (2017) as follows:

$$\begin{aligned} P_t &= p(x_{0:t}, \mathbf{C}_{1:t} \mid u_{1:t}, z_{1:t}, y_{1:t}, f_{1:t}, AM, \mathbf{h}) \\ &\approx p(i_t, C_t \mid x_{0:t}, i_{1:t-1}, C_{1:t-1}, S_{1:t}, f_{1:t}, \mathbf{h}) \\ &\quad \cdot p(z_t \mid x_t, m_{t-1}) p(f_t \mid C_{1:t-1}, f_{1:t-1}, \mathbf{h}) \\ &\quad \cdot p(x_t \mid x_{t-1}, u_t) p(S_t \mid S_{1:t-1}, y_{1:t}, AM, \lambda) \\ &\quad \cdot \underbrace{p(x_t \mid x_{0:t-1}, i_{1:t-1}, C_{1:t-1}, \mathbf{h})}_{\text{Additional part}} \\ &\quad \cdot \frac{p(S_t \mid S_{1:t-1}, C_{1:t-1}, \alpha, \beta)}{p(S_t \mid S_{1:t-1}, \beta)} \cdot P_{t-1}, \end{aligned} \tag{2}$$

where the term $p(x_t \mid x_{0:t-1}, i_{1:t-1}, C_{1:t-1}, \mathbf{h})$ is the additional part compared to the original equation.

Here, the target distribution for the particle filter is the marginal joint posterior distribution of the self-positions $x_{0:t}$ and the set of latent variables $\mathbf{C}_{1:t}$ because it is based on the RBPF technique adopted in FastSLAM in the same manner. The latent variables that are local parameters are estimated by a particle filter, and the probability distributions for global parameters LM, Θ, and m are calculated and held independently for each estimated particle.

We describe the proposal distribution Q_t as follows:

$$\begin{aligned} Q_t &= q(x_{0:t}, \mathbf{C}_{1:t} \mid u_{1:t}, z_{1:t}, y_{1:t}, f_{1:t}, AM, \mathbf{h}) \\ &= p(x_t \mid x_{t-1}, z_t, m_{t-1}, u_t) \\ &\quad \cdot p(i_t, C_t \mid x_{0:t}, i_{1:t-1}, C_{1:t-1}, S_{1:t}, f_{1:t}, \mathbf{h}) \\ &\quad \cdot p(S_t \mid S_{1:t-1}, y_{1:t}, AM, \lambda) \cdot Q_{t-1}. \end{aligned} \tag{3}$$

Then, $p(x_t \mid x_{t-1}, z_t, m_{t-1}, u_t)$ is equivalent to the proposal distribution of FastSLAM 2.0. The probability distribution of i_t and C_t is the marginal distribution pertaining to the set of model parameters Θ. This distribution can be calculated using a formula equivalent to collapsed Gibbs sampling. The details are described in Taniguchi et al. (2017).

3.2.2 Sampling of words using speech recognition and word segmentation

We approximate the probability distribution of S_t in (3) as speech recognition with the language model LM_{t-1} and unsupervised word segmentation using the WFST speech recognition results with latticelm (Neubig et al. 2012) as follows:

$$\begin{aligned} &p(S_t \mid S_{1:t-1}, y_{1:t}, AM, \lambda) \\ &\quad \approx \text{latticelm}(S_{1:t} \mid \mathcal{L}_{1:t}, \lambda) \text{SR}(\mathcal{L}_{1:t} \mid y_{1:t}, AM, LM_{t-1}) \end{aligned} \tag{4}$$

where SR() denotes the function of speech recognition, $\mathcal{L}_{1:t}$ denotes the speech recognition results in WFST format, which is a word graph representing the speech recognition results. In the original mathematical formulas, only S_t should be obtained by sampling. However, latticelm is a tool originally designed for batch learning. In addition, in order to perform unsupervised word segmentation, it is necessary to extract statistical information from the observation data. Therefore, resampling is necessary using all data from 1 to t, instead of exclusively using the distribution at time-step t.

3.2.3 Additional weight for i_t, C_t, and x_t (AW)

Finally, the importance weight ω_t modified from Taniguchi et al. (2017) is represented as follows:

$$\begin{aligned} \omega_t \approx & \underbrace{\sum_{i_t=k} \Big[p(x_t \mid x_{0:t-1}, i_{1:t-1}, i_t = k, \mathbf{h}) \cdot \sum_{C_t=l} p(i_t = k, C_t = l \mid C_{1:t-1}, i_{1:t-1}, \mathbf{h}) \Big]}_{\text{Additional part}} \\ &\cdot p(z_t \mid m_{t-1}, x_{t-1}, u_t) p(f_t \mid C_{1:t-1}, f_{1:t-1}, \mathbf{h}) \\ &\cdot \frac{p(S_t \mid S_{1:t-1}, C_{1:t-1}, \alpha, \beta)}{p(S_t \mid S_{1:t-1}, \beta)} \cdot \omega_{t-1}. \end{aligned} \tag{5}$$

Unlike the original SpCoSLAM algorithm, the marginal likelihood for i_t and C_t weighted by the marginal likelihood for the position distribution was added to the additional part of the first term on the right side of (5). The amount of calculations does not increase because most of the formulas for weight ω_t are already calculated when i_t and C_t are sampled. Weight calculation in consideration of the likelihood of the entire model can be realized by (5). This is described in Algorithm 1 (Line 16) and Algorithm 2 (Line 17).

3.2.4 Weight for selecting a language model LM (WS)

In the formulation of (1), it is desirable to estimate the language model LM_t for each particle. In other words, speech recognition of the amount of data multiplied by the number of particles for each teaching utterance must be performed. In this paper, to reduce the computational cost, we use a language model LM_t of a particle with the maximum weight for speech recognition.

We also modify the weight for selecting the language model from the entire weight ω_t of the model to the weight ω_S related to word information:

$$\omega_S = \frac{p(S_{1:t} \mid C_{1:t-1}, \alpha, \beta)}{p(S_{1:t} \mid \beta)}. \tag{6}$$

The segmentation result from all of the uttered sentences for each particle changes at every step because the word segmentation processes use all previous data. Indeed, better word segmentation results can be selected by a weight that considers not only current data but also previous data. In addition, this modified weight corresponds to mutual information used for selecting the word segmentation results in SpCoA++ (Taniguchi et al. 2018a). This is described in Algorithm 1 (Line 23) and Algorithm 2 (Line 24).

4 SpCoSLAM 2.0: improved and scalable online learning algorithm

In this section, we describe an improved and scalable online learning algorithm, SpCoSLAM 2.0, that overcomes the problems in the original algorithm. Although the generative process and graphical model for SpCoSLAM are the same, the learning algorithm is different. SpCoSLAM 2.0 is a novel learning algorithm proposed with a modified mathematical formulation that retains the model structure, similar to the extension from FastSLAM to FastSLAM 2.0. First, the algorithm is improved by introducing techniques such as rejuvenation, as explained in Sect. 4.1. Next, a scalable algorithm is developed to reduce the calculation time while maintaining higher accuracy than the original algorithm, as described in Sect. 4.2.

4.1 Improving the estimation accuracy

We now turn to the details of the improved algorithm. Here, we introduce two elements: fixed-lag rejuvenation of latent variables, and re-segmentation of word sequences. A pseudocode for the improved algorithm is given in Algorithm 1.

Algorithm 1 SpCoSLAM 2.0: Improved algorithm

```
1:  procedure SpCoSLAM2.0($X_{t-1}, u_t, z_t, f_{1:t}, y_{1:t}$)
2:      $\bar{X}_t = X_t = \emptyset$
3:      $\mathcal{L}_{1:t} = \mathrm{SR}(\mathcal{L}_{1:t} \mid y_{1:t}, AM, LM_{t-1})$
4:      for $r = 1$ to $R$ do
5:          $\acute{x}_t^{[r]} = \textbf{sample\_motion\_model}(u_t, x_{t-1}^{[r]})$
6:          $x_t^{[r]} = \textbf{scan\_matching}(z_t, \acute{x}_t^{[r]}, m_{t-1}^{[r]})$
7:          for $j = 1$ to $J$ do
8:              $x_j = \textbf{sample\_motion\_model}(u_t, x_{t-1}^{[r]})$
9:          end for
10:         $\omega_z^{[r]} = \sum_{j=1}^{J} \textbf{measurement\_model}(z_t, x_j, m_{t-1}^{[r]})$
11:         $S_{1:t}^{[r]} \sim \mathrm{latticelm}(S_{1:t} \mid \mathcal{L}_{1:t}, \lambda)$
12:         for $\tau = t - T_L + 1$ to $t$ do
13:             $i_\tau^{[r]}, C_\tau^{[r]} \sim p(i_\tau, C_\tau \mid x_{0:t}^{[r]}, S_{1:t}^{[r]}, f_{1:t},$
                        $i_{\{1:t|\neg\tau\}}^{[r]}, C_{\{1:t|\neg\tau\}}^{[r]}, \mathbf{h})$
14:         end for
15:         $\omega_f^{[r]} = p(f_t \mid C_{1:t-1}^{[r]}, f_{1:t-1}, \alpha, \chi)$
16:         $\omega_{ic}^{[r]} = \sum_{i_t=k} \Big[ p(x_t^{[r]} \mid x_{0:t-1}^{[r]}, i_{1:t-1}^{[r]}, i_t = k, \mathbf{h})$
                $\cdot \sum_{C_t=l} p(i_t = k, C_t = l \mid C_{1:t-1}^{[r]}, i_{1:t-1}^{[r]}, \mathbf{h}) \Big]$
17:         $\omega_s^{[r]} = \frac{p(S_t^{[r]} \mid S_{1:t-1}^{[r]}, C_{1:t-1}^{[r]}, \alpha, \beta)}{p(S_t^{[r]} \mid S_{1:t-1}^{[r]}, \beta)}$
18:         $\omega_t^{[r]} = \omega_z^{[r]} \cdot \omega_f^{[r]} \cdot \omega_s^{[r]} \cdot \omega_{ic}^{[r]}$
19:         $m_t^{[r]} = \textbf{updated\_occupancy\_grid}(z_t, x_t^{[r]}, m_{t-1}^{[r]})$
20:         $\Theta_t^{[r]} = E[p(\Theta \mid x_{0:t}^{[r]}, \mathbf{C}_{1:t}^{[r]}, f_{1:t}, \mathbf{h})]$
21:         $\bar{X}_t = \bar{X}_t \cup \langle x_{0:t}^{[r]}, \mathbf{C}_{1:t}^{[r]}, m_t^{[r]}, \Theta_t^{[r]}, \omega_t^{[r]} \rangle$
22:     end for
23:     $S_{1:t}^{*} = \mathrm{argmax}_{S_{1:t}^{[r]}} \sum_{r=1}^{R} \omega_S^{[r]} \delta(S_{1:t} - S_{1:t}^{[r]})$
24:     $LM_t \sim \mathrm{NPYLM}(LM \mid S_{1:t}^{*}, \lambda)$
25:     for $r = 1$ to $R$ do
26:         draw $i$ with probability $\propto \omega_t^{[i]}$
27:         add $\langle x_{0:t}^{[i]}, \mathbf{C}_{1:t}^{[i]}, m_t^{[i]}, \Theta_t^{[i]}, LM_t \rangle$ to $X_t$
28:     end for
29:     return $X_t$
30: end procedure
```

4.1.1 Fixed-lag rejuvenation of i_t and C_t (FLR–i_t, C_t)

Canini et al. (2009) demonstrated improved accuracy with rejuvenation by resampling previous samples randomly. This is based on a result of the independent and identically distributed (i.i.d.) assumption on the latent variables in the Latent Dirichlet Allocation (LDA) model. However, in the case of selecting from previous data of all time points, all previous samples need to be held in the memory. In the proposed algorithm, we introduce Fixed-Lag Rejuvenation (FLR) inspired by the Monte Carlo fixed-lag smoother (Kitagawa 2014). This approach is similar to the sampling strategy of fixed-lag roughening for particle filter-based SLAM in Beevers and Huang (2007). Beevers and Huang (2007) indicated that the statistical estimation error could be reduced by applying Markov Chain Monte Carlo (MCMC)–based sampling to the trajectory samples over a fixed lag at each time step.

The fixed-lag smoother is a particle smoothing method that estimates particles approximating the smoothing distribution $p(\mathbf{C}_\tau \mid D_{1:t})$ $(\tau < t)$, where D is observed data. It is obtained by a simple modification to the particle filter. In

Algorithm 2 SpCoSLAM 2.0: Scalable algorithm

```
procedure SpCoSLAM2.0($X_{t-1}, u_t, z_t, f_{t'+1:t}, y_{t'+1:t}$)
    $\bar{X}_t = X_t = \emptyset$
    $t' = t - T_L$
    $\mathcal{L}_{t'+1:t} = \mathrm{SR}(\mathcal{L}_{t'+1:t} \mid y_{t'+1:t}, AM, LM_{t'})$
    for $r = 1$ to $R$ do
        $\acute{x}_t^{[r]} = \textbf{sample\_motion\_model}(u_t, x_{t-1}^{[r]})$
        $x_t^{[r]} = \textbf{scan\_matching}(z_t, \acute{x}_t^{[r]}, m_{t-1}^{[r]})$
        for $j = 1$ to $J$ do
            $x_j = \textbf{sample\_motion\_model}(u_t, x_{t-1}^{[r]})$
        end for
        $\omega_z^{[r]} = \sum_{j=1}^{J} \textbf{measurement\_model}(z_t, x_j, m_{t-1}^{[r]})$
        $S_{t'+1:t}^{[r]} \sim \mathrm{latticelm}(S_{t'+1:t} \mid \mathcal{L}_{t'+1:t}, \lambda)$
        for $\tau = t' + 1$ to $t$ do
            $i_\tau^{[r]}, C_\tau^{[r]} \sim p(i_\tau, C_\tau \mid x_{t'+1:t}^{[r]}, S_{t+1:t}^{[r]}, f_{t'+1:t},$
                    $i_{\{t'+1:t|\neg\tau\}}^{[r]}, C_{\{t'+1:t|\neg\tau\}}^{[r]}, H_{t'}^{[r]})$
        end for
        $\omega_f^{[r]} = p(f_t \mid C_{t'+1:t-1}^{[r]}, f_{t'+1:t-1}, H_{t'}^{[r]})$
        $\omega_{ic}^{[r]} = \sum_{i_t=k} \Big[ p(x_t^{[r]} \mid x_{t'+1:t-1}^{[r]}, i_{t'+1:t-1}^{[r]},$
                    $i_t = k, H_{t'}^{[r]}) \sum_{C_t=l} p(i_t = k, C_t = l \mid$
                    $C_{t'+1:t-1}^{[r]}, i_{t'+1:t-1}^{[r]}, H_{t'}^{[r]}) \Big]$
        $\omega_s^{[r]} = \frac{p(S_t^{[r]} \mid S_{t'+1:t-1}^{[r]}, C_{t'+1:t-1}^{[r]}, H_{t'}^{[r]})}{p(S_t^{[r]} \mid S_{t'+1:t-1}^{[r]}, H_{t'}^{[r]})}$
        $\omega_t^{[r]} = \omega_z^{[r]} \cdot \omega_f^{[r]} \cdot \omega_s^{[r]} \cdot \omega_{ic}^{[r]}$
        $m_t^{[r]} = \textbf{updated\_occupancy\_grid}(z_t, x_t^{[r]}, m_{t-1}^{[r]})$
        $H_t^{[r]} = F[p(\Theta \mid x_{t'+1:t}^{[r]}, \mathbf{C}_{t'+1:t}^{[r]}, f_{t'+1:t}, H_{t'}^{[r]})]$
        $\bar{X}_t = \bar{X}_t \cup \langle x_{t'+1:t}^{[r]}, \mathbf{C}_{t'+1:t}^{[r]}, m_t^{[r]}, H_{t'+1:t}^{[r]}, \omega_t^{[r]} \rangle$
    end for
    $S_{t'+1:t}^{*} = \mathrm{argmax}_{S_{t'+1:t}^{[r]}} \sum_{r=1}^{R} \omega_S^{[r]} \delta(S_{t'+1:t} - S_{t'+1:t}^{[r]})$
    $LM_t = \mathrm{argmax}_{LM}\, p(LM \mid S_{t'+1:t}^{*}, LM_{t'}, \lambda)$
    for $r = 1$ to $R$ do
        draw $i$ with probability $\propto \omega_t^{[i]}$
        add $\langle x_{t'+1:t}^{[i]}, \mathbf{C}_{t'+1:t}^{[i]}, m_t^{[i]}, H_{t'+1:t}^{[i]}, LM_{t'+1:t} \rangle$ to $X_t$
    end for
    return $X_t$
end procedure
```

this algorithm, particles are saved from time-step $t - T_L + 1$ to t and are resampled according to the weight based on newly observed data each step. Here, the value of the fixed-lag is denoted by T_L. This technique means that the particles at step τ can be estimated not by using the observed data $D_{1:\tau}$, but rather with $D_{1:\tau+T_L}$, i.e., the smoothing distribution $p(\mathbf{C}_\tau \mid D_{1:\tau+T_L})$. In general, a smoothing method such as a fixed-lag particle smoother provides more accurate estimations than naive online estimation methods such as a particle filter in estimating the joint posterior distribution of latent variables.

Figure 3 shows an overview of the FLR of i_t and C_t. The notation $\tau \mid t$ in the box in Fig. 3 is shorthand notation for the subscript representing the time-step in the conditional marginal posterior distribution, e.g., $p(\mathbf{C}_\tau \mid D_{1:t})$. The FLR is the process of sampling the latent variables i_τ and C_τ by iterating T_L times from the previous step $t - T_L + 1$ to the current step t for each particle as follows:

$$i_\tau, C_\tau \sim p(i_\tau, C_\tau \mid x_{0:t}, S_{1:t}, f_{1:t}, i_{\{1:t|\neg\tau\}}, C_{\{1:t|\neg\tau\}}, \mathbf{h}) \tag{7}$$

where $i_{\{1:t|\neg\tau\}}$ and $C_{\{1:t|\neg\tau\}}$ denote sets of elements from 1 to t without the elements of step τ. In this case, the latent variables of step $t - T_L$ can be sampled using data up to step t, as described in Algorithm 1 (Lines 12–14). Equation (7) is the same as the conditional posterior probability distribution for marginalized (collapsed) Gibbs sampling used in batch learning. Therefore, the FLR corresponds to slightly iterate Gibbs sampling for some recent previous latent variables in online learning.

4.1.2 Re-segmentation of word sequences (RS)

We introduce re-segmentation of word sequences to improve the accuracy of word segmentation. In the original algorithm, we approximated the left side of (4) by registering the word sequences segmented by latticelm to the word dictionary. However, this can be considered a process of sampling a language model LM from word sequences $S_{1:t}^{*}$ and a hyperparameter λ of a language model. Therefore, we adopt NPYLM, an unsupervised word segmentation method (Mochihashi et al. 2009), to estimate a language model from the word sequences as follows:

$$LM \sim \mathrm{NPYLM}(LM \mid S_{1:t}^{*}, \lambda). \tag{8}$$

The procedure of introducing the RS is as follows: (i) word sequences $S_{1:t}$ are obtained by WFST speech recognition and latticelm; (ii) word sentences $S_{1:t}^{*}$ of a maximum likelihood particle are converted into syllable sequences, and segmented into word sequences using NPYLM; (iii) the word dictionary LM is updated using segmented words, as described in Algorithm 1 (Line 24). In this manner, we can overcome problematic words that tend to become under-segmented while taking into account the uncertainty of speech recognition errors by latticelm. Note that there is a discrepancy between the words used for spatial concept acquisition and the word set registered in the word dictionary.

4.2 Scalability for reduced computational cost

In this section, we describe the details of the scalable algorithm. Here, we introduce two elements: the sequential Bayesian update of the parameters in the posterior distribution, and unsupervised word segmentation from WFST speech recognition results using FLR. The scalable algorithm

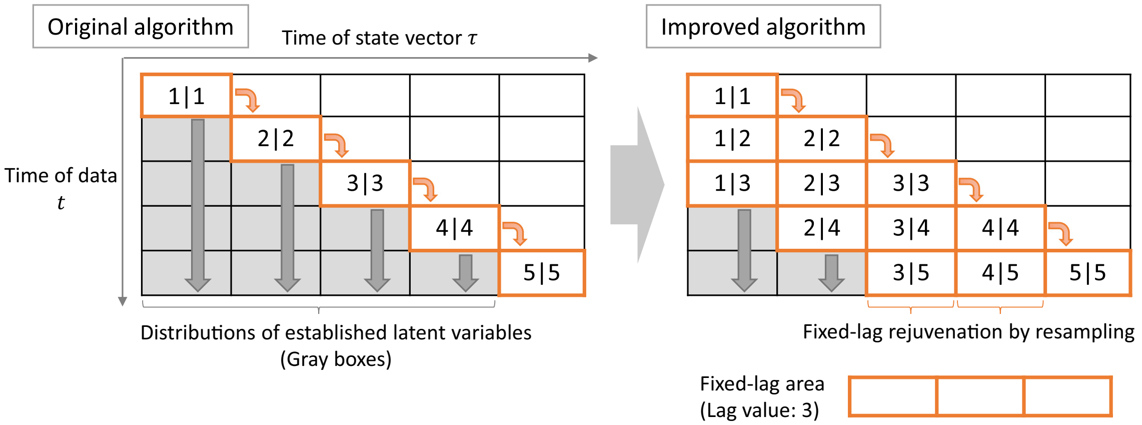


Fig. 3 Overview of the Fixed-Lag Rejuvenation of i_t and C_t. Left: naive online learning in the original algorithm. Right: online learning using FLR in the improved algorithm. The thick orange frame is estimated by sampling. In this case, the fixed-lag value T_L is three. The gray boxes mean that the estimated value will never again be updated, i.e., distributions of already immobilized (fixed) latent variables by online learning (Color figure online)

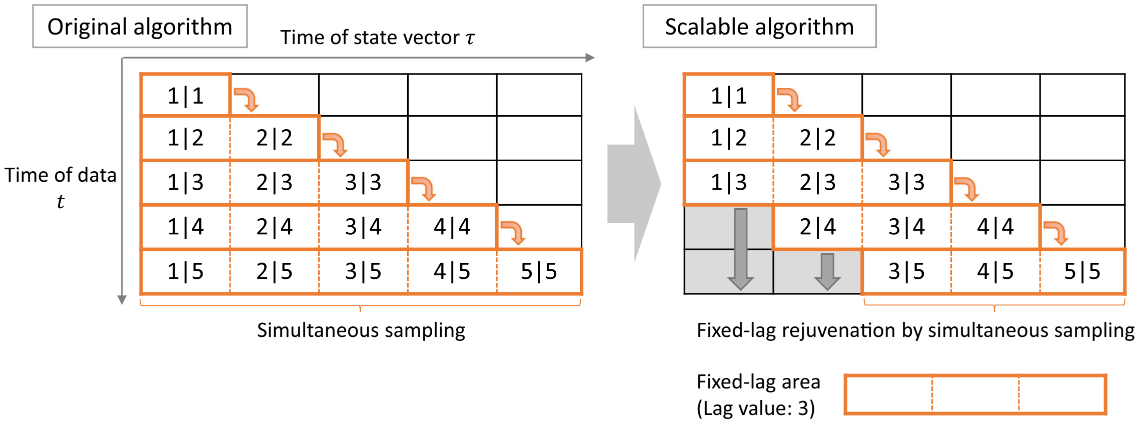


Fig. 4 Overview of the fixed-lag rejuvenation of S_t. Left: batch learning with the original algorithm. Right: pseudo-online learning using FLR in the scalable algorithm. The thick orange frame is estimated by sampling from the joint distribution. In this case, the fixed-lag value T_L is three. The gray boxes denote that the estimated value will never be updated again, i.e., distributions of already immobilized (fixed) latent variables by online learning (Color figure online)

can be combined with the FLR C_t, i_t of the improved algorithm. The pseudo-code for the scalable algorithm is given in Algorithm 2.

4.2.1 Sequential Bayesian update of parameters in the posterior distribution (SBU)

We introduce a Sequential Bayesian Update (SBU) for the posterior hyperparameters H_t in the posterior distribution. In the original algorithm, the model parameters Θ are estimated from all the data $D_{1:t} = \{f_{1:t}, y_{1:t}\}$ and the set of latent variables $\mathbf{C}_{1:t}$ during each step. However, FastSLAM avoids holding all the previous data by updating a map m_t from x_t, z_t, and m_{t-1} sequentially. That is, it assumes the measurement model $p(z_t \mid x_{0:t}, z_{1:t-1}) = p(z_t \mid x_t, m_{t-1})$ and the updated occupancy grid map $p(m_t \mid x_{0:t}, z_{1:t}) = p(m_t \mid x_t, z_t, m_{t-1})$. Similarly, the posterior hyperparameters H_t can be calculated from the new data D_t, latent variables $\mathbf{C}_t$, and posterior hyperparameters H_{t-1} from previous steps. Thus, both the computational and memory efficiency, crucial for long-term learning with real robots, can be significantly improved. The SBU for the posterior hyperparameters is calculated as follows:

$$\begin{aligned} p(\Theta \mid H_t) &= p(\Theta \mid D_{1:t}, \mathbf{C}_{1:t}, \mathbf{h}) \\ &= p(\Theta \mid D_t, \mathbf{C}_t, \{D_{1:t-1}, \mathbf{C}_{1:t-1}, \mathbf{h}\}) \\ &= p(\Theta \mid D_t, \mathbf{C}_t, H_{t-1}) \\ &\propto p(D_t \mid \mathbf{C}_t, \Theta) p(\Theta \mid H_{t-1}). \end{aligned} \tag{9}$$

These posterior hyperparameters H_t can also be used to sample $\mathbf{C}_t$. In the implementation, it suffices to hold values of the statistics obtained during the calculation of the posterior distribution. Here, the calculation results from the left side and the right side of (9) are strictly the same. The SBU approach is also said to keep track of sufficient statistics in the particle filter (Kantas et al. 2015).

The SBU equation is used together with FLR as follows:

$$\begin{aligned} p(\Theta \mid H_t) &= p(\Theta \mid D_{1:t}, \mathbf{C}_{1:t}, \mathbf{h}) \\ &= p(\Theta \mid D_{t'+1:t}, \mathbf{C}_{t'+1:t}, \{D_{1:t'}, \mathbf{C}_{1:t'}, \mathbf{h}\}) \\ &= p(\Theta \mid D_{t'+1:t}, \mathbf{C}_{t'+1:t}, H_{t'}) \\ &\propto p(D_{t'+1:t} \mid \mathbf{C}_{t'+1:t}, \Theta) p(\Theta \mid H_{t'}), \end{aligned} \tag{10}$$

where a time-step before the lag value is $t' = t - T_L$. In this case, it is only necessary to hold the observed data and posterior hyperparameters of the number corresponding to the lag value T_L. Equation (10) is applied to Algorithm 2.

4.2.2 WFST speech recognition and unsupervised word segmentation using FLR (FLR–S_t)

We describe the proposed algorithm that combines FLR and SBU to address problems of the unsupervised online word segmentation and to reduce the computation time simultaneously. FLR can also be extended to the sampling of S_t in a pseudo-online manner. Figure 4 shows an overview of the FLR of S_t. The notation $\tau \mid t$ takes the same meaning as it does in Fig. 3. The data used for speech recognition and word segmentation is modified from that in (4) to data with a fixed-lag interval. In addition, speech recognition is performed using the initial syllable dictionary in the steps before step T_L and using a word dictionary from step t' in the steps proceeding step $T_L + 1$. In this case, we can perform word segmentation based on the statistical information collected from the WFSTs recognized using the number of data for the lag value T_L. FLR performs simultaneous sampling of word sequences $S_{t'+1:t}$ of time-steps from $t' + 1$ to the current step t as follows:

$$\begin{aligned} S_{t'+1:t} &\sim p(S_{t'+1:t} \mid y_{t'+1:t}, AM, S_{1:t'}, \lambda) \\ &\approx \text{latticelm}(S_{t'+1:t} \mid \mathcal{L}_{t'+1:t}, \lambda) \\ &\quad \cdot \text{SR}(\mathcal{L}_{t'+1:t} \mid y_{t'+1:t}, AM, LM_{t'}). \end{aligned} \tag{11}$$

Therefore, this approach can address the problem in the original algorithm by which incorrect word segmentation in early learning stages was propagated to the following learning stages.

Here, the amount of calculations is constant throughout each step, irrespective of the total amount of data. This property of the FLR of S_t is an important advantage in scalability. However, there is a concern that word segmentation using FLR becomes inaccurate compared to batch learning because of the limited availability of statistical information. Essentially, the scalable algorithm is a trade-off between calculation time and word segmentation accuracy. In the language model update, the word dictionary LM_t holds information regarding words $S_{t'+1:t}$ segmented from steps $t' + 1$ to t and the previous word dictionary $LM_{t'}$. This is described in Algorithm 2 (Lines 4, 12, and 25).

Table 1 Computational complexity of the learning algorithms

Algorithm	Order
SpCpSLAM (Taniguchi et al. 2017)	$O(NR)$
SpCoSLAM 2.0 (Improved)	$O(NR)$
SpCoSLAM 2.0 (Scalable)	$O(T_L R)$
SpCoA (Taniguchi et al. 2016) (Batch learning)	$O(NG)$
SpCoA++ (Taniguchi et al. 2018a) (Batch learning)	$O(NGMI)$

Table 1 shows the order of computational complexity for each learning algorithm. The data number is denoted N, the number of particles R, the value of fixed-lag T_L, the number of iterations for Gibbs sampling in batch learning G, the number of candidates of word segmentation results for updating the language model in SpCoA++ M, and the number of iterations for the parameter estimation in SpCoA++ I. Variables without N are constants that can be preset by the user. Among these algorithms, therefore, only the scalable algorithm does not depend on the number of data N. In this case, the computational efficiency of the scalable algorithm is better than the original SpCoSLAM algorithm when $T_L < N$.

5 Experiment I

We performed experiments to demonstrate online learning of spatial concepts in a novel environment. In addition, we performed evaluations of place categorization and lexical acquisition related to places. We compared the performance of the following methods:

(A) SpCoSLAM (Taniguchi et al. 2017)
(B) SpCoSLAM with AW + WS (Sect. 3.2)
(C) SpCoSLAM 2.0 (FLR–i_t, C_t)
(D) SpCoSLAM 2.0 (FLR–i_t, C_t + RS)
(E) SpCoSLAM 2.0 (FLR–i_t, C_t, S_t + SBU)
(F) SpCoA++ (Batch learning) (Taniguchi et al. 2018a)

Methods (A) and (B) used the original and modified SpCoSLAM algorithms. Methods (C) and (D) used the proposed improved algorithms under different conditions. In methods (C) and (D), the lag value for FLR was set to $T_L =$

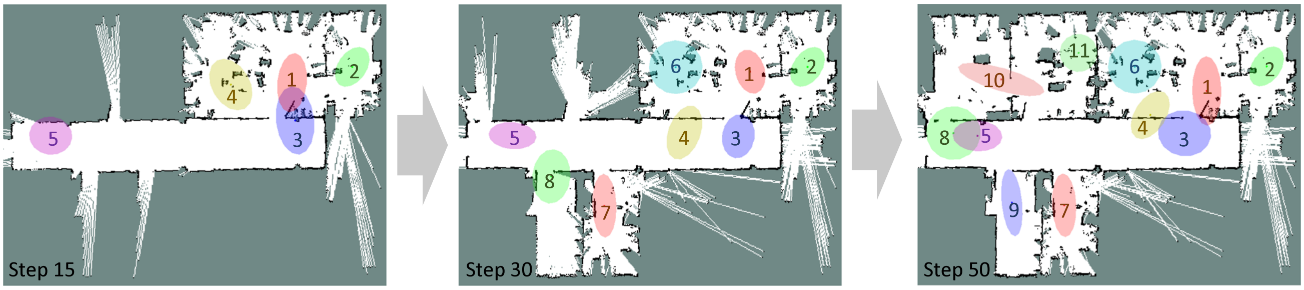


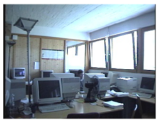
Image	Correct word (*English*)	i_t	Step 15	i_t	Step 30	i_t	Step 50
	/kyouyuuseki/ (*Shared desk*)	1	/nobasyanyanamae/ /bawa**kyoo**/ /**yuseki**/	1	/**yuseki**nikibashita/ /**kyoyuseki**/ /**kyoyuseki**dayo/	1	/kibashita/ /ni/ /**kyouyuseki**/
	/ikidomari/ (*End of the corridor*)	3	/namaewa/ /enikimashita/ /gonobashiha/	3	/dayo/ /**ikidomari**/ /fokoga/	3	/**idomari**/ /miriiNgusupesu/ /koko/
	/roboqtookiba/ (*Robot storage space*)	4	/**robotokiba**/ /**robotokiba**nya/ /rimasu/	6	/kochirawaaga**ro**/ /**botoki**/ /baninarimasu/	6	/**wabotookiba**/ /**robotookiba**/ /kochiraga/

Fig. 5 Top: learning results of position distributions in a generated map. Ellipses denote the position distributions drawn on the map at steps 15, 30, and 50. The colors of the ellipses were randomly determined for each index number $i_t = k$. Bottom: examples of scene images captured by the robot. The correct word (in English) and estimated words are shown for each position distribution at steps 15, 30, and 50 (Color figure online)

10. Method (E) used the proposed scalable algorithm under three different conditions: the lag values for the FLR were set to $T_L = 1$, 10, and 20 for (E1), (E2), and (E3), respectively. Batch-learning methods (F) was estimated by Gibbs sampling based on a weak-limit approximation (Fox et al. 2011) of the Stick-Breaking Process (SBP) (Sethuraman 1994), one of the constitutive methods of the Dirichlet Process (DP). The upper limits of the spatial concepts and position distributions were set to $L = 50$ and $K = 50$, respectively. We set the number of iterations for Gibbs sampling to $G = 100$. In method (F), we set the number of candidate word segmentation results for updating the language model to $M = 6$, and the number of iterative estimation procedures to $I = 10$. In addition, (F) did not use image features in the same manner as the original model setting. Note that SpCoA++ (F) was not evaluated in Taniguchi et al. (2017) because it is the latest batch-learning method.

5.1 Online learning

We conducted experiments of online spatial concept acquisition in a real environment. We implemented SpCoSLAM 2.0 based on the open-source SpCoSLAM[1], extending the gmapping package and implementing grid-based FastSLAM 2.0 (Grisetti et al. 2007) in the Robot Operating System (ROS). We used an open dataset, albert-b-laser-vision, i.e., a rosbag file containing the odometry, laser range data, and image data. This dataset was obtained from the Robotics Data Set Repository (Radish) (Howard and Roy 2003). We prepared Japanese speech data corresponding to the movement of the robot from the above-mentioned dataset because speech data was not initially included. The total number of taught utterances was $N = 50$, including 10 types of phrases. The robot learned 10 places and 9 place names. The microphone was a SHURE PG27-USB. Julius dictation-kit-v4.4 (DNN-HMM decoding) (Lee and Kawahara 2009) was used as a speech recognizer. The initial word dictionary contained 115 Japanese syllables. The unsupervised word segmentation system used latticelm (Neubig et al. 2012). The image feature extractor was implemented with Caffe, a deep-learning framework (Jia et al. 2014). We used a pre-trained CNN model, Places365-ResNet, trained with 365 scene categories from the Places2 Database with 1.8 million images (Zhou

[1] https://github.com/a-taniguchi/SpCoSLAM2.

et al. 2018). The number of particles was $R = 30$. The hyperparameters for online learning were set as follows: $\alpha = 20$, $\gamma = 0.1$, $\beta = 0.1$, $\chi = 0.1$, $m_0 = [0, 0]^{\mathrm{T}}$, $\kappa_0 = 0.001$, $V_0 = \mathrm{diag}(2, 2)$, and $\nu_0 = 3$. The above-mentioned parameters were set such that all online methods were tested under the same conditions. The hyperparameters for batch learning were set as follows: $\alpha = 10, \gamma = 10, \beta = 0.1, m_0 = [0, 0]^{\mathrm{T}}$, $\kappa_0 = 0.001$, $V_0 = \mathrm{diag}(2, 2)$, and $\nu_0 = 3$. The hyperparameters were determined manually and empirically according to each method. Note that the speech recognition decoder, the image feature extractor, and the hyperparameters were changed from Taniguchi et al. (2017).

Figure 5 (top) shows the position distributions in the environmental maps at steps 15, 30, and 50 with (D). This figure visualizes how spatial concepts are acquired during sequential mapping of the environment. The position distributions were appropriately formed for places uttered by a user each time. In step 15, the map covers only 2 rooms (in the upper right) and a corridor, with 5 position distributions. The map obtained at step 50 covers the entire environment, and there were eventually 11 estimated position distributions. Figure 5 (bottom) shows an example of the correct phoneme sequence of the place name, and the three best words estimated by the probability distribution $p(S_t \mid i_t, \Theta_t, LM_t)$ at step t. The left side shows an example of the scene images observed in the i_t-th position distribution corresponding to the name of each place. As the steps proceed, it can be seen that the words corresponding to the places were stably learned as phoneme sequences closer to the correct answers. For example, in /kyouyuuseki/ (*shared desk*), in step 15, the correspondence between the place and phoneme sequence was insufficiently learned: e.g., /bawakyoo/ and /yuseki/. However, by step 50, the word was learned correctly: /kyouyuseki/. The index of the position distribution of /roboqtookiba/ (*robot storage space*) was changed from 4 to 6. This change means that the label number switched as a result of the previous estimate values being modified while learning progressed. Details of the online learning experiment can be found in a video online[2].

5.2 Evaluation metrics

We evaluated the different algorithms according to the following metrics: the Adjusted Rand Index (ARI) (Hubert and Arabie 1985) of the classification results of spatial concepts $C_{1:N}$ and position distribution $i_{1:N}$; the Estimation Accuracy Rate (EAR) of the estimated total numbers of spatial concepts L and position distributions K; and the Phoneme Accuracy Rate (PAR) of uttered sentences and words related to places. We conducted six learning trials under each algorithm condition. The details of the evaluation metrics are described in the following sections.

5.2.1 Estimation accuracy of spatial concepts

We compared the matching rate for the estimated indices $C_{1:N}$ of the spatial concept and the classification results of the correct answers given by a person. In this experiment, the evaluation metric adopts the ARI, which is a measure of the similarity between two clustering results. The matching rate for the estimated indices $i_{1:N}$ of the position distributions was evaluated in the same manner.

In addition, we evaluated the estimated number of spatial concepts L and position distributions K using the EAR. The EAR was calculated as follows:

$$\mathrm{EAR} = \max\left(1 - \frac{\mid n_t^{\mathrm{C}} - n_t^{\mathrm{E}} \mid}{n_t^{\mathrm{C}}}, 0\right) \tag{12}$$

where n_t^{C} is the correct number and n_t^{E} is the estimated number at time-step t.

5.2.2 PAR of uttered sentences

We next compared the accuracy rate of phoneme recognition and word segmentation for all the recognized sentences. However, it was difficult to separately weigh the ambiguous phoneme recognition and the unsupervised word segmentation. Therefore, the experiment considered the position of a delimiter as a single letter. The correct phoneme sequence was suitably segmented into Japanese morphemes using MeCab (Kudo 2006), an off-the-shelf Japanese morphological analyzer that is widely used for natural language processing. However, the name of the place was considered a single word.

We calculated the PAR of the uttered sentences with the correct phoneme sequence s_t^{P}, and a phoneme sequence s_t^{R} of the recognition result of each uttered sentence. The PAR was calculated as follows:

$$\mathrm{PAR} = \max\left(1 - \frac{\mathrm{LD}(s_t^{\mathrm{P}}, s_t^{\mathrm{R}})}{n^{\mathrm{P}}}, 0\right) \tag{13}$$

where LD() was calculated using the Levenshtein distance between s_t^{P} and s_t^{R}. Here, n^{P} denotes the number of phonemes of the correct phoneme sequence.

5.2.3 PAR of words related to places

We also evaluated whether a phoneme sequence has learned the properly segmented place names. This experiment assumed a request for the best phoneme sequence, s_t^*, representing the self-position x_t of the robot. We compared the PAR of words with the correct place name and a selected word for each teaching place. The PAR was calculated using (13).

[2] https://youtu.be/H5yztfmxGbc

Table 2 Evaluation results in a real environment

Metric		Improved	Scalable	ARI		EAR		PAR	
				C_t	i_t	L	K	Sentence	Word
(A)	SpCoSLAM			0.273	0.502	0.756	0.881	0.524	0.154
(B)	SpCoSLAM with AW + WS			0.233	0.420	0.805	0.901	0.496	0.086
(C)	SpCoSLAM 2.0 (10 FLR–i_t, C_t)	✓		0.324	0.602	0.876	0.913	0.533	0.157
(D)	SpCoSLAM 2.0 (10 FLR–i_t, C_t + RS)	✓		0.320	0.555	**0.881**	0.901	**0.801**	0.419
(E1)	SpCoSLAM 2.0 (1 FLR–i_t, C_t, S_t + SBU)	✓	✓	0.244	0.443	0.869	**0.923**	0.648	0.158
(E2)	SpCoSLAM 2.0 (10 FLR–i_t, C_t, S_t + SBU)	✓	✓	0.314	0.570	0.790	0.801	0.690	0.262
(E3)	SpCoSLAM 2.0 (20 FLR–i_t, C_t, S_t + SBU)	✓	✓	0.351	**0.673**	0.748	0.890	0.704	0.292
(F)	SpCoA++ (Batch learning)			**0.387**	0.624	0.700	0.648	0.787	**0.524**

Bold underlined indicate the highest evaluation values, and underline indicates the second highest evaluation values

The selection of a word $s^*_{t,b}$ was calculated as follows:

$$s^*_t = \mathrm{argmax}_{S_{t,b}}\ p(S_{t,b} \mid x_t, \Theta_t, LM_t). \qquad (14)$$

In this experiment, we used the self-position x_t that was not included in the training data to evaluate the PAR of words. Here, the robot can perform sufficiently accurate self-localization using a laser range finder. Therefore, in this experiment, we assume that x_t is given an accurate coordinate value without errors.

The more a method accurately recognized words and acquired spatial concepts, the higher is the PAR. We consider this evaluation metric to be an overall measure of the proposed method.

5.3 Evaluation results and discussion

In this section, we discuss the improvement and scalability of the proposed learning algorithms. Table 2 lists the averages of the evaluation values calculated using the metrics ARI, EAR, and PAR at step 50.

5.3.1 ARI and EAR results

In terms of categorization accuracy, the proposed algorithms that introduced FLR tended to show higher ARI values than the original algorithms (A) and (B) of SpCoSLAM. Figure 6 shows examples of the progress of place clustering for position distributions in (A) and (D). The step numbers in the figures on the left (A) and right (D) are not the same. In these cases, large position distributions covering distant areas were learned, i.e., the purple ellipses in the figures on top. In (A), incorrect clustering results were obtained during the final step (i.e., step 50) because the original SpCoSLAM algorithm cannot correct past erroneous estimations. By contrast, in (D) by introducing FLR, an incorrect cluster occurred at step 25 (top right figure). However, the proposed algorithm could correct previous erroneous estimates at step 30 (bottom right figure). Therefore, in the original algorithm (A), estimation errors adversely affect subsequent estimations. However, SpCoSLAM 2.0 (D) obtained more accurate estimations immediately, despite previous incorrect estimations. Similar situations to (D) were also confirmed in other proposed algorithms that introduced FLR. Experimental results demonstrated that FLR, which resamples the latent variables of the previous step using observations up to the current step, contributes to improving the accuracy of online place clustering.

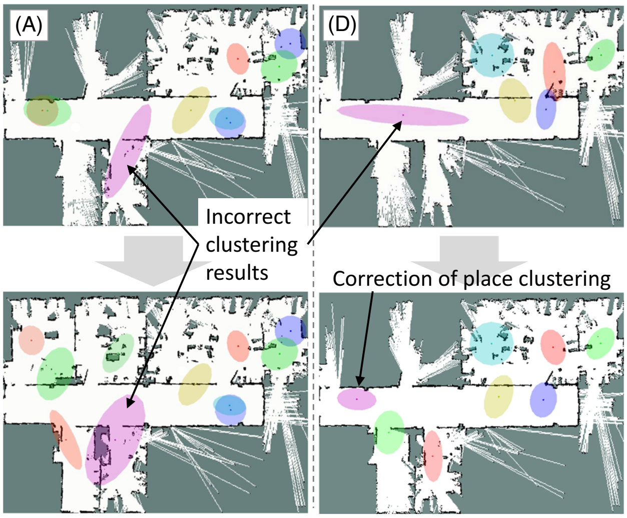


Fig. 6 Examples of corrected place clustering results. Left: the original algorithm (A). Right: the improved SpCoSLAM 2.0 algorithm (D)

Figure 7 shows the results of the EAR values with spatial concepts and position distributions, i.e., the accuracy of the

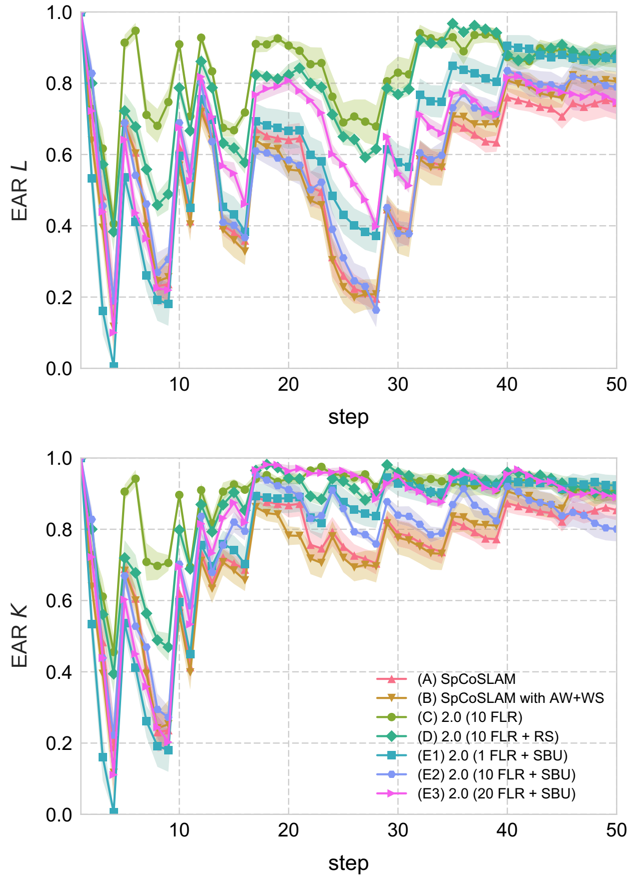


Fig. 7 Change in the EAR regarding the estimated total number of spatial concepts L (top) and position distributions K (bottom) for each step

estimated number of clusters, for each step. The EAR values were not stable in the steps during the first half, although they converged stably to high values in the latter half. In the result at step 50, (D) showed the highest EAR value L and (E1) showed the highest EAR value K. However, for both L and K, looking at all the steps on average, (A) and (B) yielded relatively low values overall, and (C) and (D) yielded relatively high values. (E1)–(E3) tended to show values between original algorithms, (A) and (B), and improved algorithms with FLR, (C) and (D). From the results of (C) and (D), EAR values improved considerably by introducing the FLR of C_t and i_t.

5.3.2 PAR sentence and word results

From the results of the improved algorithm (D), the PAR values (sentence and word) improved markedly by adding the re-segmentation of the word sequences. These results show that the robot can accurately segment the names of places and learn the relationship between places and words more precisely. In particular, method (D), which combines the FLR and RS, achieved an overall improvement comparable to the other online algorithms. Some trial results showed PAR values comparable to those of SpCoA++ (F). Figure 8 shows the PAR of words for each step. The PAR tended to increase as a whole. Therefore, it can be expected that the PAR values will further increase as the number of steps advances. Table 3 presents examples of word segmentation results with the four methods. The correct phoneme sequence, i.e., ground truth, was segmented into Japanese morphemes using MeCab (Kudo 2006), where "|" denotes a delimiter, i.e., a word segment position. The parts in bold correspond to the name for each place. SpCoSLAM (A) showed under-segmentation results in many cases. On the other hand, it can be seen that SpCoSLAM 2.0 (D) and (E3) properly segmented the phoneme sequences representing the name of the place. Comparing (D) and (E3), (D) obtained segmentation results close to those of the batch learning method (F), and (E3) sometimes slightly over-segmented words. Therefore, SpCoSLAM 2.0 can mitigate under-segmentation when the word segmentation of the batch learning method is applied in a pseudo-online manner.

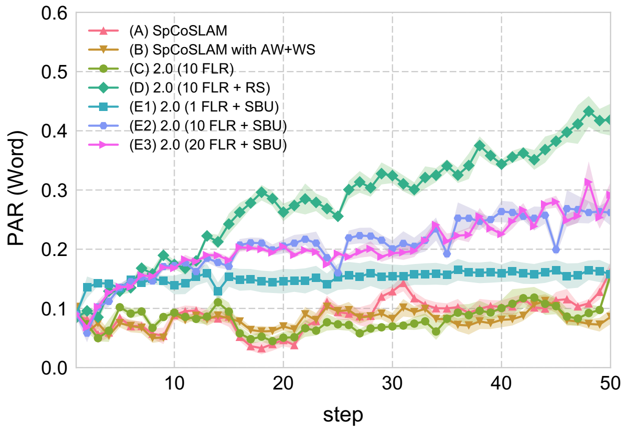


Fig. 8 Change in the PAR of words for each step

5.3.3 Original and modified SpCoSLAM algorithms

Although the modified SpCoSLAM (B) is theoretically more appropriate than the original algorithm (A), few differences were found between them. In the proposed algorithms, the time-driven process, i.e., SLAM part, and the event-driven process, i.e., spatial concept formation and lexical acquisition, were estimated by the same particle filter. Although self-localization and mapping were performed each time the robot moved in an environment, latent variables for the spatial concepts and lexicon are updated only upon the user's utterance. Thus, particles can fluctuate as a result of resampling due to movement in the absence of the user's utterance. Consequently, the weight for self-localization might be influ-

Table 3 Examples of word segmentation results of uttered sentences

English	“*This place is* **the shared desk**.”
Ground truth	kochira | ga | **kyouyuuseki** | ni | nari | masu
(A)	a | kochiraga**gyoyusekiN**ni | narimasu
(D)	kochira | ga | **kyouyuseki** | ninarimasu
(E3)	uo | kochi | ra | ga | **kyoyuseki** | nina | ri | ma | su
(F)	ochiraga | **kyoyuseki** | ninarimasu
English	“*This is* **the meeting space**.”
Ground truth	koko | wa | **miitiNgusupeisu** | desu
(A)	kokowaga | **midigisupesu**desujoouya
(D)	kokowa | **miriiNgusupesu** | desu
(E3)	kowa | **midigyusu** | **pesu** | desu
(F)	gokoga | **miidiNgusupesu** | desu
English	“**The printer room** *is here.*”
Ground truth	**puriNtaabeya** | wa | kochira | desu
(A)	io**poriNtabea**akochiragadesuduuryuzu qaqo
(D)	**puriNtabeya** | kochira | desu
(E3)	**puriNpabeya** | ta | kochiradesu
(F)	**poriNpabeya** | wakochiradesu

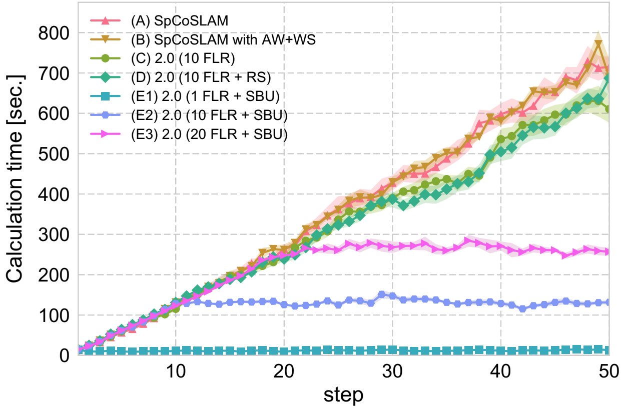


Fig. 9 Calculation times par step for evaluating scalability

ential, rather than the weight for the spatial concept and lexicon. This will be investigated in future work.

5.3.4 Calculation time and scalable algorithm

Figure 9 shows the calculation times between online learning algorithms. With batch learning, SpCoA++'s overall calculation time including the runtime of rosbag for SLAM was 13,850.873 s, and the calculation times per iteration for the iterative estimation procedure and Gibbs sampling were 1,318.954 s and 1.833 s, respectively. In the original SpCoSLAM algorithm, (A) and (B), and the improved SpCoSLAM 2.0 algorithm, (C) and (D), the calculation time increased with the number of steps, i.e., as the amount of data increased. However, the scalable SpCoSLAM 2.0 algorithm (E1)–(E3) retained a constant calculation time regardless of an increase in the amount of data. Therefore, we can exert particularly powerful effects for long-term learning.

In the scalable algorithm (E1)–(E3), the evaluation values of ARI and PAR tended to improve overall when the lag value increased. In particular, when the lag value was 20, relatively high evaluation values are seen to approach those of the improved algorithm.

Owing to a trade-off between the fixed-lag size and accuracy, the algorithm needs to be set appropriately according to both the computational power embedded in the robot and the duration requirements for actual operation. In this experiment, we did not evaluate the scalability of the algorithm with parallel processing. However, we considered that the proposed algorithm could be executed even faster by parallelizing the particle process and by using Graphics Processing Units (GPUs). As such, we consider that the robot would be able to move within the environment while learning in real-time.

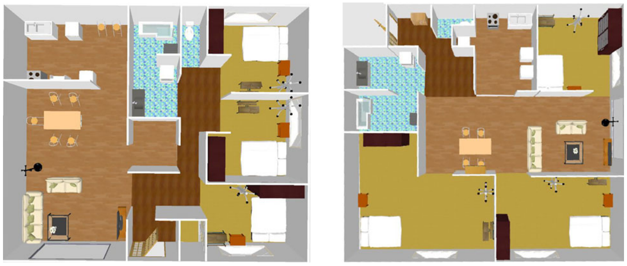


Fig. 10 Examples of home environments in SIGVerse

6 Experiment II

In this experiment, it is investigated whether trends similar to the evaluation results of the real environmental dataset in Sect. 5 can be stably obtained across different environments. Place categorization and lexical acquisition related to places in virtual home environments were evaluated, and the evaluation metrics ARI, EAR, and PAR for the methods (A)–(F) were compared in the same manner as in Sect. 5.

6.1 Condition

Online spatial concept acquisition experiments were conducted in various virtual home environments. The simulator environment was SIGVerse version 3.0 (Inamura et al. 2010), a client-server based architecture that can connect the ROS and Unity. The virtual robot in SIGVerse was Toyota's Human Support Robot (HSR), and we used 10 different

Table 4 Evaluation results in simulator environments

Metric		ARI		EAR		PAR	
		C_t	i_t	L	K	Sentence	Word
(A)	SpCoSLAM	0.252	0.604	0.785	0.818	0.558	0.098
(B)	SpCoSLAM with AW + WS	0.347	0.684	0.802	0.815	0.565	0.141
(C)	SpCoSLAM 2.0 (10 FLR–i_t, C_t)	0.346	0.713	0.733	0.868	0.553	0.096
(D)	SpCoSLAM 2.0 (10 FLR–i_t, C_t + RS)	0.314	0.719	0.730	0.840	**0.835**	0.464
(E1)	SpCoSLAM 2.0 (1 FLR–i_t, C_t, S_t + SBU)	0.307	0.672	0.817	0.800	0.671	0.165
(E2)	SpCoSLAM 2.0 (10 FLR–i_t, C_t, S_t + SBU)	0.385	0.688	0.833	0.782	0.733	0.305
(E3)	SpCoSLAM 2.0 (20 FLR–i_t, C_t, S_t + SBU)	0.354	0.790	**0.883**	**0.898**	0.768	0.350
(F)	SpCoA++ (Batch learning)	**0.522**	**0.899**	0.800	0.850	0.830	**0.480**

Bold underlined indicate the highest evaluation values, and underline indicates the second highest evaluation values

home environments[3] created using Sweet Home 3D[4], which is a free software for interior design application. Figure 10 shows examples of the home environments. For each place, 10 training data were provided on average. The total number of taught utterances was $N = 60$, including 10 types of phrases. The robot learned six places and their respective names. The microphone and speech recognizer were the same as those in Sect. 5.1. The image feature extractor was a pre-trained BVLC CaffeNet model (Jia et al. 2014). The number of particles was $R = 10$. The hyperparameters for learning were set as follows: $\alpha = 10.0$, $\gamma = 1.0$, $\beta = 0.1$, $\chi = 0.1$, $m_0 = [0, 0]^{\mathrm{T}}$, $\kappa_0 = 0.001$, $V_0 = \mathrm{diag}(2, 2)$, and $\nu_0 = 3$. The hyperparameters were determined manually and empirically. The above-mentioned parameters were set such that all methods were tested under the same conditions. In method (F), the upper limits of the spatial concepts and position distributions were set to $L = 20$ and $K = 20$, respectively. The other settings were identical to those in Sect. 5.

The main target of the evaluation in this study is the accuracy of place clustering and lexical acquisition, i.e., extended points in SpCoSLAM 2.0. Therefore, in this experiment, it is assumed that sufficiently accurate mapping and self-localization are possible with a high-precision distance sensor, and using an online learning algorithm which separates and omits the SLAM process was executed. The true values obtained by the simulator were used as the self-position data.

6.2 Result

In this section, the improvement and scalability of the proposed learning algorithms in home environments are discussed. Table 4 lists the averages of the evaluation values calculated using the metrics ARI, EAR, and PAR at step 60.

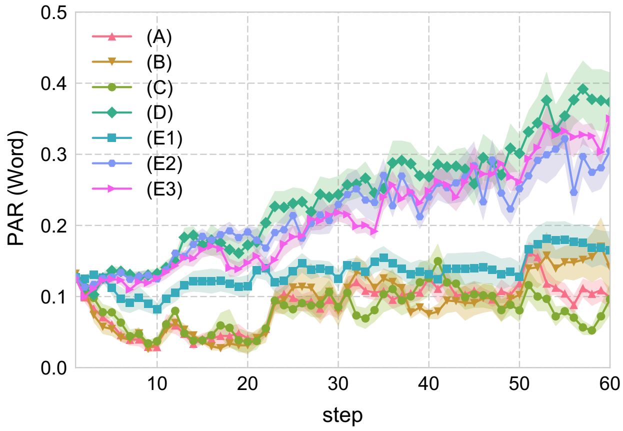


Fig. 11 Change in PAR of words for each step in simulator environments

The ARI showed a similar trend as the result of real environmental data. However, compared to the original algorithms (A) and (B), there was almost no difference in the values in algorithms that introduced FLR. In addition, the EAR showed a slightly different trend than the real environmental data. In the improved algorithms (C) and (D), the number L of categories of spatial concepts smaller than the true value was estimated compared to other algorithms. We consider that this reason was due to the fact that it was re-combined into the same category by FLR. Because the dataset was obtained in the simulator environment, for example, the image features could be insufficiently obtained for place categorization, i.e., similar features might be found in different places. Such a problem did not occur when using real environmental data.

The PAR had the same tendency as the result of real environment data. Similar to Sect. 5.3, the improved algorithm with RS (D) showed lexical acquisition accuracy comparable to batch learning (F). In addition, the scalable algorithms with FLR of S_t (E2) and (E3) showed higher values than the original algorithms. Figure 11 shows the average values of

[3] 3D models of home environments are available in https://github.com/a-taniguchi/SweetHome3D_rooms.

[4] Sweet Home 3D: http://www.sweethome3d.com/

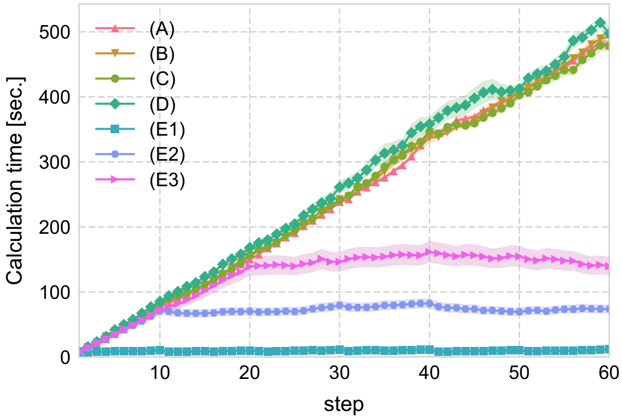


Fig. 12 Calculation times par step in simulator environments

the PAR of words for each step in different environments. Similar to Fig. 8, the PAR tended to increase overall. Thus, it can be seen that RS and FLR of S_t work effectively in virtual home environments.

In the comparison of the original and modified SpCoSLAM algorithms (A) and (B), the modified algorithm (B) showed higher overall values in the evaluation values of ARI and PAR. We consider that the weight for the spatial concept and lexicon acted more directly in this experiment than in the experiment in Sect. 5, because it was not affected by the weight for self-localization.

In scalable algorithms (E1)–(E3), as the FLR value increased, the tendency for the overall evaluation values to increase appeared more prominently than for the results of real environment data.

Figure 12 shows the average calculation times between online learning algorithms in simulator environments. We confirmed that the result was similar to Fig. 9, which was the result using the real environment data. With batch learning, SpCoA++'s overall average calculation time was 8,076.288 s, and the calculation times per iteration for the iterative estimation procedure and Gibbs sampling were 807.623 s and 1.346 s, respectively.

The following are the common inferences from the results of both the simulation and real-world environments. For the online learning, if the user requires the performance of lexical acquisition even at an increased time cost, they can execute the improved algorithm (D) or scalable algorithm with a larger lag value, e.g., (E2) and (E3). If the user requires high-speed calculation, they can obtain better results faster than the conventional algorithm (A) by executing a scalable algorithm such as (E1) and (E2).

7 Conclusion

This paper proposed an improved and scalable online learning algorithm to address the problems encountered by our previously proposed SpCoSLAM algorithm. Specifically, we proposed online learning algorithm, called SpCoSLAM 2.0, for spatial concepts and lexical acquisition, for higher accuracy and scalability. In experiments, we conducted online learning with a robot in a novel environment without any pre-existing lexicon and map. In addition, we compared the proposed algorithm to the original online algorithm and to batch learning in terms of the estimation accuracy and calculation time. The results demonstrate that the proposed algorithm is more accurate than the original algorithm and of comparable accuracy to batch learning. Moreover, the calculation time of the proposed scalable algorithm becomes constant for each step, regardless of the amount of training data. We expect this work to contribute to the realization of long-term spatial language interactions between humans and robots.

In the future, we shall experiment with long-term online learning of spatial concepts in large-scale environments based on the scalable algorithm proposed in this paper. Furthermore, with additional development, it will be possible to introduce a forgetting mechanism to the proposed algorithm as with Araki et al. (2012a). When a robot continues to operate over a long period of time it will encounter changes in the environment, such as the names of places and areas. Consequently, the robot will benefit from using the latest observation data as opposed to the previous observation data. We believe that such a mechanism will be especially effective for long-term learning.

The proposed method constructs spatial concepts on a metric map; however, it can also be extended to learning the topological structure of places as with Karaoğuz and Bozma (2016); Luperto and Amigoni (2018). We explore whether this facilitates navigation tasks with human–robot linguistic interactions. In addition, loop-closure detection has been studied actively in recent years, as is evident from long-term visual SLAM (Han et al. 2018). The generative model of SpCoSLAM is connected to SLAM and lexical acquisition via latent variables related to the spatial concepts. Therefore, we shall also explore loop-closure detection based on speech signals and investigate whether spatial concepts can positively affect mapping.

We will explore whether the SpCoSLAM model proposed herein can be integrated with other probabilistic models to form a large-scale cognitive model for general-purpose autonomous intelligent robots using a SERKET architecture (Nakamura et al. 2018). However, applications of the SERKET architecture are limited due to its computational cost for learning the enormous parameters of the whole model. Even in such a case, we consider that our proposed approach to online learning will be extensively useful because it can be applied to various other Bayesian models.

Acknowledgements The authors thank Cyrill Stachniss for providing the dataset in the real robot. The authors also thank Kazuya Asada and Keishiro Taguchi for providing virtual home environments and the dataset for spatial concepts in SIGVerse simulator.

References

Aldous, D. (1985). Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII-1983* (pp. 1–198).

Aoki, T., Nishihara, J., Nakamura, T., & Nagai, T. (2016). Online joint learning of object concepts and language model using multimodal hierarchical Dirichlet process. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 2636–2642). IEEE

Araki, T., Nakamura, T., Nagai, T., Funakoshi, K., Nakano, M., & Iwahashi, N. (2012a). Online object categorization using multimodal information autonomously acquired by a mobile robot. *Advanced Robotics*, *26*(17), 1995–2020.

Araki, T., Nakamura, T., Nagai, T., Nagasaka, S., Taniguchi, T., & Iwahashi, N. (2012b). Online learning of concepts and words using multimodal LDA and hierarchical Pitman-Yor Language Model. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1623–1630). IEEE

Ball, D., Heath, S., Wiles, J., Wyeth, G., Corke, P., & Milford, M. (2013). OpenRatSLAM: an open source brain-based slam system. *Autonomous Robots*, *34*(3), 149–176.

Beevers, K. R., & Huang, W. H. (2007). Fixed-lag sampling strategies for particle filtering slam. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 2433–2438). IEEE

Börschinger, B., & Johnson, M. (2011). A particle filter algorithm for Bayesian wordsegmentation. In *Australasian language technology association workshop 2011* (p. 10). Citeseer

Börschinger, B., & Johnson, M. (2012). Using rejuvenation to improve particle filtering for Bayesian word segmentation. In *Proceedings of the 50th annual meeting of the association for computational linguistics, association for computational linguistics* (pp. 85–89).

Cangelosi, A., & Schlesinger, M. (2015). Developmental robotics: From babies to robots. intelligent robotics and autonomous agents series. MIT Press. https://books.google.co.jp/books?id=AbKPoAEACAAJ.

Canini, K. R., Shi, L., & Griffiths, T. L. (2009). Online inference of topics with latent Dirichlet allocation. *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, *9*, 65–72.

Doucet, A., De Freitas, N., Murphy, K., & Russell, S. (2000). Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the 16th conference on uncertainty in artificial intelligence* (pp. 176–183). Morgan Kaufmann Publishers Inc.

Fox, E. B., Sudderth, E. B., Jordan, M. I., & Willsky, A. S. (2011). A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics*, *5*(2A), 1020–1056.

Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, *23*, 34–46.

Gu, Z., Taguchi, R., Hattori, K., Hoguro, M., & Umezaki, T. (2016). Learning of relative spatial concepts from ambiguous instructions. In *Proceedings of the 13th IFAC/IFIP/IFORS/IEA symposium on analysis, design, and evaluation of human-machine systems (IFAC HMS)* (Vol. 49, pp. 150–153). Elsevier

Hagiwara, Y., Inoue, M., Kobayashi, H., & Taniguchi, T. (2018). Hierarchical spatial concept formation based on multimodal information for human support robots. *Frontiers in Neurorobotics*, *12*, 11. https://doi.org/10.3389/fnbot.2018.00011.

Han, F., Wang, H., Huang, G., & Zhang, H. (2018). Sequence-based sparse optimization methods for long-term loop closure detection in visual slam. *Autonomous Robots*, *42*(7), 1323–1335. https://doi.org/10.1007/s10514-018-9736-3.

Heath, S., Ball, D., & Wiles, J. (2016). Lingodroids: Cross-situational learning for episodic elements. *IEEE Transactions on Cognitive and Developmental Systems*, *8*(1), 3–14. https://doi.org/10.1109/TAMD.2015.2442619.

Hemachandra, S., Walter, M. R., Tellex, S., & Teller, S. (2014). Learning spatial-semantic representations from natural language descriptions and scene classifications. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 2623–2630). IEEE

Howard, A., & Roy, N. (2003). The robotics data set repository (radish). http://radish.sourceforge.net/.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Inamura, T., Shibata, T., Sena, H., Hashimoto, T., Kawai, N., Miyashita, T., Sakurai, Y., Shimizu, M., Otake, M., Hosoda, K., et al. (2010). Simulator platform that enables social interaction simulation—SIGVerse: SocioIntelliGenesis simulator. In: *Proceedings of the IEEE/SICE international symposium on system integration* (pp. 212–217).

Isobe, S., Taniguchi, A., Hagiwara, Y., & Taniguchi, T. (2017). Learning relationships between objects and places by multimodal spatial concept with bag of objects. In *Proceedings of the international conference on social robotics (ICSR)* (pp. 115–125). Springer

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). On particle methods for parameter estimation in state-space models. *Statistical Science*, *30*(3), 328–351.

Karaoğuz, H., & Bozma, H. I. (2016). An integrated model of autonomous topological spatial cognition. *Autonomous Robots*, *40*(8), 1379–1402. https://doi.org/10.1007/s10514-015-9514-4.

Kitagawa, G. (2014). Computational aspects of sequential Monte Carlo filter and smoother. *Annals of the Institute of Statistical Mathematics*, *66*(3), 443–471.

Kostavelis, I., & Gasteratos, A. (2015). Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, *66*, 86–103.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the advances in neural information processing systems (NIPS)*, Nevada, United States (pp. 1097–1105).

Kudo, T. (2006). MeCab: Yet another part-of-speech and morphological analyzer. https://github.com/taku910/mecab.

Landsiedel, C., Rieser, V., Walter, M., & Wollherr, D. (2017). A review of spatial reasoning and interaction for real-world robotics. *Advanced Robotics*, *31*(5), 222–242.

Lee, A., & Kawahara, T. (2009). Recent development of open-source speech recognition engine Julius. In *Proceedings of the APSIPA ASC* (pp. 131–137).

Luperto, M., & Amigoni, F. (2018). Predicting the global structure of indoor environments: A constructive machine learning approach. *Autonomous Robots*. https://doi.org/10.1007/s10514-018-9732-7.

Mochihashi, D., Yamada, T., & Ueda, N. (2009). Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP (ACL-IJCNLP)* (pp. 100–108).

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the international joint conference on artificial intelligence (IJCAI)* (pp. 1151–1156).

Nakamura, T., Nagai, T., & Taniguchi, T. (2018). Serket: An architecture for connecting stochastic models to realize a large-scale cognitive model. *Frontiers in Neurorobotics*, *12*, 25. https://doi.org/10.3389/fnbot.2018.00025.

Neubig, G., Mimura, M., & Kawahara, T. (2012). Bayesian learning of a language model from continuous speech. *IEICE Transactions on Information and Systems*, *95*(2), 614–625.

Nishihara, J., Nakamura, T., & Nagai, T. (2017). Online algorithm for robots to learn object concepts and language model. *IEEE Transactions on Cognitive and Developmental Systems*, *9*(3), 255–268. https://doi.org/10.1109/TCDS.2016.2552579.

Pronobis, A., & Jensfelt, P. (2012). Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 3515–3522). IEEE

Rangel, J. C., Cazorla, M., García-Varea, I., Romero-González, C., & Martínez-Gómez, J. (2018). Automatic semantic maps generation from lexical annotations. *Autonomous Robots*. https://doi.org/10.1007/s10514-018-9723-8.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, *4*, 639–650.

Sünderhauf, N., Dayoub, F., McMahon, S., Talbot, B., Schulz, R., Corke, P., Wyeth, G., Upcroft, B., & Milford, M. (2016). Place categorization and semantic mapping on a mobile robot. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 5729–5736). IEEE

Taguchi, R., Yamada, Y., Hattori, K., Umezaki, T., Hoguro, M., Iwahashi, N., Funakoshi, K., & Nakano, M. (2011). Learning place-names from spoken utterances and localization results by mobile robot. In *Proceedings of the annual conference of the international speech communication association (INTERSPEECH)* (pp. 1325–1328).

Taniguchi, A., Taniguchi, T., & Inamura, T. (2016). Spatial concept acquisition for a mobile robot that integrates self-localization and unsupervised word discovery from spoken sentences. *IEEE Transactions on Cognitive and Developmental Systems*, *8*(4), 285–297. https://doi.org/10.1109/TCDS.2016.2565542.

Taniguchi, A., Hagiwara, Y., Taniguchi, T., & Inamura, T. (2017). Online spatial concept and lexical acquisition with simultaneous localization and mapping. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 811–818). https://doi.org/10.1109/IROS.2017.8202243.

Taniguchi, A., Taniguchi, T., & Inamura, T. (2018a). Unsupervised spatial lexical acquisition by updating a language model with place clues. *Robotics and Autonomous Systems*, *99*, 166–180. https://doi.org/10.1016/j.robot.2017.10.013.

Taniguchi, T., Ugur, E., Hoffmann, M., Jamone, L., Nagai, T., Rosman, B., et al. (2018b). Symbol emergence in cognitive developmental systems: a survey. *IEEE transactions on cognitive and developmental systems* (pp. 1–1). https://doi.org/10.1109/TCDS.2018.2867772.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. Cambridge: MIT Press.

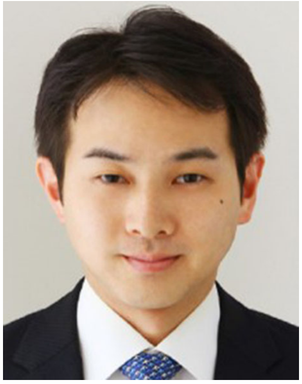
Ueda, R., Mizuta, K., Yamakawa, H., & Okada, H. (2016). Particle filter on episode for learning decision making rule. In *Proceedings of the international conference on intelligent autonomous systems (IAS)* (pp. 737–754). Springer

Walter, M.R., Hemachandra, S., Homberg, B., Tellex, S., & Teller, S. (2013). Learning semantic maps from natural language descriptions. In *Proceedings of robotics: science and systems (RSS)*.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2018). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(6), 1452–1464.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Akira Taniguchi received his B.E., M.E., and Ph.D. degree from Ritsumeikan University, Kyoto, Japan, in 2013, 2015, and 2018, respectively. From April 2017 to March 2018, he was a research fellow of Japan Society for the Promotion of Science (DC2). From April 2018 to March 2019, he was a research fellow of Japan Society for the Promotion of Science (PD). From April 2019, he is currently a Specially Appointed Assistant Professor at the College of Information Science and Engineering, Ritsumeikan University. His research interests include intelligent robotics, artificial intelligence, and symbol emergence in robotics.



Yoshinobu Hagiwara received his Ph.D. degree from Soka University, Japan, in 2010. He was an Assistant Professor at the Department of Information Systems Science, Soka University from 2010, a Specially Appointed Researcher at the Principles of Informatics Research Division, National Institute of Informatics from 2013, and an Assistant Professor at the Department of Human and Computer Intelligence, Ritsumeikan University from 2015. He is currently a Lecture at the Department of Information Science and Engineering, Ritsumeikan University. His research interests include human-robot interaction, machine learning, intelligent robotics and symbol emergence in robotics. He is a member of IEEE, RSJ, IEEJ, JSAI, SICE, and IEICE.

Tadahiro Taniguchi received M.E., and Ph.D. degrees from Kyoto University in 2003 and 2006, respectively. From April 2005 to March 2008, he was a research fellow of Japan Society for the Promotion of Science. He was an assistant professor from April 2008 to March 2010, an associate professor from April 2010 to March 2017. He has been a professor in the College of Information Science and Engineering, Ritsumeikan University since April 2017, and a visiting general chief scientist in Panasonic since April 2017. From September 2015 to September 2016, he was a visiting associate professor at Imperial College London. He is currently engaged in research on artificial intelligence, symbol emergence in robotics and emergent systems.

Tetsunari Inamura received B.E., M.S., and PhD. degrees from the University of Tokyo in 1995, 1997, and 2000, respectively. He was a Researcher of the JST/CREST program from 2000 to 2003, and then joined the Department of Mechano-Informatics, School of Information Science and Technology, the University of Tokyo as a Lecturer till 2006. He is now an Associate Professor in National Institute of Informatics and the Department of Informatics, SOKENDAI (The Graduate University for Advanced Studies). His research interests include imitation learning, human motion analysis, and development of interactive robots through virtual reality.