Author version. KI 2018, LNAI 11117, pp. 117-130, 2018. The final authenticated publication is available online at https://doi.org/10.1007/978-3-030-00111-7_11

A Sequence-Based Neuronal Model for Mobile Robot Localization

Peer Neubert[1,2], Subutai Ahmad[2], and Peter Protzel[1]

[1] Chemnitz University of Technology, 09126 Chemnitz, Germany
[2] Numenta, Inc., Redwood City, CA, United States
peer.neubert@etit.tu-chemnitz.de

Abstract. Inferring ego position by recognizing previously seen places in the world is an essential capability for autonomous mobile systems. Recent advances have addressed increasingly challenging recognition problems, e.g. long-term vision-based localization despite severe appearance changes induced by changing illumination, weather or season. Since robots typically move continuously through an environment, there is high correlation within consecutive sensory inputs and across similar trajectories. Exploiting this sequential information is a key element of some of the most successful approaches for place recognition in changing environments. We present a novel, neurally inspired approach that uses sequences for mobile robot localization. It builds upon Hierarchical Temporal Memory (HTM), an established neuroscientific model of working principles of the human neocortex. HTM features two properties that are interesting for place recognition applications: (1) It relies on sparse distributed representations, which are known to have high representational capacity and high robustness towards noise. (2) It heavily exploits the sequential structure of incoming sensory data. In this paper, we discuss the importance of sequence information for mobile robot localization, we provide an introduction to HTM, and discuss theoretical analogies between the problem of place recognition and HTM. We then present a novel approach, applying a modified version of HTM's higher order sequence memory to mobile robot localization. Finally we demonstrate the capabilities of the proposed approach on a set of simulation-based experiments.

Keywords: Mobile Robot Localization · Hierarchical Temporal Memory · Sequence-based Localization.

1 Introduction

We describe the application of a biologically detailed model of sequence memory in the human neocortex to mobile robot localization. The goal is to exploit the sequence processing capabilities of the neuronal model and its powerful sparse distributed representations to address particularly challenging localization tasks. Mobile robot localization is the task of determining the current position of the robot relative to its own prior experience or an external reference frame (e.g. a map). Due to its fundamental importance for any robot aiming at performing meaningful tasks, mobile robot localization is a long studied problem, going back to visual landmark-based navigation in Shakey the robot in the 1960-80s [23]. Research has progressed rapidly over the last few decades and it has become possible to address increasingly challenging localization tasks. The problem of localization in the context of changing environments, e.g. recognizing a cloudy winter scene which has been seen previously on a sunny summer day, has only recently been studied [17,19]. In most applications, the robot's location changes smoothly and there are no sudden jumps to other places (the famous kidnapped robot problem appears only rarely in practice [7]). Therefore a key element of some of the most successful approaches is to exploit the temporal consistency of observations.

In this paper, we present a localization approach that takes inspiration from sequence processing in Hierarchical Temporal Memory (HTM) [11,6,9], a model of working principles of the human neocortex. The underlying assumption in HTM is that there is a single cortical learning algorithm that is applied everywhere in the neocortex. Two fundamental working principles of this algorithm are to learn from sequences to predict future neuronal activations and to use sparse distributed representations (SDRs). In Section 2 we first provide a short overview of recent methods to exploit sequential information for robot localization. In Section 3 we provide an overview of the HTM sequence memory algorithm. In section 4 we show how HTM's higher order sequence memory can be applied to the task of mobile robot place recognition[3]. We identify a weakness of the existing HTM approach for place localization and discuss an extension of the original algorithm. We discuss theoretical analogies of HTM and the problem of place recognition, and finally provide initial experimental results on simulated data in section 5.

2 On the Importance of Sequences for Robot Localization

Mobile robot localization comprises different tasks, ranging from recognizing an already visited place to simultaneously creating a map of an unknown area while localizing in this map (known as SLAM). The former task is known as place recognition problem or loop closure detection. A survey is provided in [15]. A solution to this problem is fundamental for solving the full SLAM problem. The research progress in this area recently reached a level where it is feasible to think about place recognition in environments with significantly changing appearances. For example, camera based place recognition under changing lighting condition, changing weather, and even across different seasons [17,19]. In individual camera images of a scene, the appearance changes can be tremendous. In our own prior work and others, the usage of sophisticated landmark detectors and deep-learning-based descriptors showed to be a partial solution of this task [21]. However, with increasing severity of the appearance changes, making the localization decision purely based on individual images is more and more pushed to its limits.

[3] An open source implementation is available: `https://www.tu-chemnitz.de/etit/proaut/seqloc`

The benefit of exploiting sequence information is well accepted in the literature [17,5,13,14,18,16]. In 2012, Milford et al.[17] presented a simple yet effective way to exploit the sequential character of the percepts of the environment. Given two sequences of images, captured during two traversals through the same environment, the task is to make a decision, which image pairs show the same place. In their experiments one sequence is from a sunny summer day and the other from a stormy winter night. To address this challenging problem, the pairwise similarity of images from the two runs is collected in a matrix. Instead of evaluating each entry individually, Milford et al. [17] propose to search for linear segments of high similarity in this matrix (this also involves a local contrast normalization). This approach significantly improved the state of the art at this time. However, searching for linear segments in this matrix poses important limitations on the data: the data on both environmental traverses has to be captured at the same number of frames per traveled distance. This is usually violated in practice, e.g., if the vehicle's velocity changes. Therefore, several extensions have been proposed. E.g., allowing non-zero acceleration [14] or searching for optimal paths in the similarity matrix using a graph-theoretical max-flow formulation [18]. Localization approaches that include the creation of a map inherently exploit the sequential nature of the data. Simultaneous creation of a map while localizing in this map exploits sequence information by creating a prior for the current position based on the previous data. However, this is equivalent to solving the full SLAM problem and involves maintaining a map of the environments. A particular challenge for SLAM are the consistency of the map after closing long loops and the increasing size and complexity of the map in large environments. One elegant approach to the latter problem is RatSLAM [16]; it uses a finite space representation to encode the pose in an infinite world. The idea is inspired by entorhinal grid cells in the rat's brain. They encode poses similar to a residual number system in math by using the same representatives (i.e. cells) for multiple places in the world. In RatSLAM, grid cells are implemented in form of a three dimensional continuous attractor network (CAN) with wrap-around connections; one dimension for each degree of freedom of the robot. The activity in the CAN is moved based on proprioceptive clues of the robot (e.g. wheel encoders) and new energy is injected by connections from local view cells that encode the current visual input, as well as from previously created experiences. The dynamics of the CAN apply a temporal filter on the sensory data. Only in case of repeated consistent evidence for recognition of a previously seen place, this matching is also established in the CAN representation. Although the complexity and number of parameters of this system prevented a wider application, RatSLAM's exploitation of sequence information allowed to demonstrate impressive navigation results.

3 Introduction to HTM

Hierarchical Temporal Memory (HTM) [9] is a model of working principles of the human neocortex. It builds upon the assumption of a single learning algorithm that is deployed all over the neocortex. The basic theoretical framework builds upon Jeff Hawkins' book from 2004 [10]. It is continuously evolving, with the goal to explain more and more aspects of the neocortex as well as extending the range of practical demonstrations and applications. Currently, these applications include anomaly detection, natural language processing and, very recently, object detection [12]. A well maintained implementation is available [2].

Although the system is continuously evolving, there is a set of entrenched fundamental concepts. Two of them are (1) the exploitation of sequence information and (2) the usage of Sparse Distributed Representations (SDRs). The potential benefit of the first concept for mobile robot localization has been elaborated in the previous section. The latter concept, SDRs, also showed to be beneficial in various fields. A SDR is a high dimensional binary vector (e.g. 2,048 dimensional) with very few 1-bits (e.g. 2%). There is evidence that SDRs are a widely used representation in brains due to their representation capacity, robustness to noise and power efficiency [3]. They are a special case of hypervector encodings, which we previously used to learn simple robot behavior by imitation learning [22].

From HTM, we want to exploit the concept of higher order sequence memory for our localization task. It builds on a set of neuronal cells with connection and activation patterns that are closer to the biological paragon than, e.g., a multi-layer perceptron or a convolutional neural network. Nevertheless, for these structures, there are compact and clear algorithmic implementations.

3.1 Mimicking neuroanatomic structures

The anatomy of the neocortex obeys a regular structure with several horizontal layers, each composed by vertically arranged minicolumns with multiple cells. In HTM, each cell incorporates dendritic properties of pyramidal cells [25]. Feed-forward inputs (e.g. perception clues) are integrated through proximal dendrites. Basal and apical dendrites provide feedback modulatory input. Feed-forward input can activate cells and modulatory input can predict activations of cells. Physiologically, predicted cells are depolarized and fire sooner than non-depolarized cells. Modulatory dendrites consist of multiple segments. Each segment can connect to a different set of cells and responds to an individual activation pattern. The dendrite becomes active if any of its segments is active. All cells in a minicolumn share the same feed-forward input, thus all cells in an minicolumn become potentially active if the feed-forward connections perceive a matching input pattern. From these potentially active cells, the actual active cells (coined winner cells) are selected based on the modulatory connections. In HTM theory, the modulatory connections provide context information for the current feed-forward input. At each timestep, multiple cells in multiple minicolumns are active and the state of the system is represented by this sparse code. For description of HTM theory and current developments please refer to [10] and [1].

3.2 Simplified Higher Order Sequence Memory (SHOSM)

In the following, we will give details on a particular algorithm from HTM: higher order sequence memory [11,6]. We will explain a simplified version that we abbreviate SHOSM. For those who are familiar with HTM: the simplifications include the absence of a spatial pooler and segments, the usage of one-shot learning instead of Hebbian-like learning, and SHOSM does not start from a randomly initialized set of minicolumns (whose connections are adapted) but starts from an empty set of minicolumns and increases the number of minicolumns on demand. Goal of the higher order sequence memory is to process an incoming sensor data stream in a way that similar input sequences create similar representations within the network - this matches very well to the sequence-based localization problem formulation. The listing in algorithm 1 describes the operations:

Algorithm 1: SHOSM - Simplified HTM higher order sequence memory

Data: I^t the current input; M a potentially empty set of existing minicolumns; C^{t-1}_{winner} the set of winner cells from the previous time step
Result: M with updated states of all cells; C^t_{winner}

$M^t_{active} = match(I^t, M)$ `// Find the active minicolumns based on similarity to feed-forward SDR input`

`// If there are no similar minicolumns: create new minicolumns`
if *isempty*(M^t_{active}) **then**
  $M^t_{active} = createMinicolumns(I^t)$ `// Each new minicolumn samples connections to 1-bits in` I^t
  $M = M \cup M^t_{active}$

`// Identify winner cell(s) in each minicolumn based on predictions`
foreach $m \in M^t_{active}$ **do**
  $C^t_{predicted} = getPredictedCells(m)$ `// Get set of predicted cells from this active minicolumn` m
  $M = activatePredictions(C^t_{predicted})$ `// Predict for next timestep`
  C^t_{winner} += $C^t_{predicted}$ `// The predicted cells are also winner cells`

  `// If there are no predicted cells: burst and select new winner`
  if *isempty*($C^t_{predicted}$) **then**
    $M = activatePredictions(m)$ `// Bursting: Activate all predictions of cells in` m `for next timestep`
    C^t_{winner} += $selectWinner(m)$ `// Select cell with the fewest predictive forward connections as winner cell`

  `// Learn predictions: prev. winner cells shall predict current`
  foreach $c \in C_{winner}$ **do**
    $learnConnections(c, C^{t-1}_{winner})$ `// Given the current winning cell` c `and the set of previously winning cells` C^{t-1}_{winner}`: for all cells` $c^{t-1}_{winner} \in C^{t-1}_{winner}$ `for which there is not already a connection from their minicolumns to the cell` c`, create the prediction connections` $c^{t-1}_{winner} \rightarrow c$ `(one shot learning)`

At each timestep, input is an SDR encoding of the current input (e.g. the current camera image). For details on SDRs and possible encodings please refer to [3] and [24]. Please keep in mind that all internal representations in algorithm 1 are SDRs: there are always multiple cells from multiple minicolumns active in parallel. Although the same input is represented by multiple minicolumns, each minicolumn connects only to a fraction of the dimensions of the input SDR and is thus affected differently by noise or errors in the input data. The noise robustness of this system is a statistical property of the underlying SDR representation [3].

In each iteration of SHOSM, a sparse set of winner cells based on the feed-forward SDR input and modulatory input from the previous iteration is computed (lines 8 and 11). Further, the *predicted* attribute of cells is updated to provide the modulatory input for the next iteration (lines 7 and 10). This modulatory prediction is the key element to represent sequences. In case of no predicted cells in an active minicolumn (line 9), all cells activate their predictions and a single winner cell is selected (this mechanism is called *bursting*). This corresponds to current input data that has never been seen in this sequence context before.

This short description of the algorithm lacks many implementation details, e.g. how exactly the connections are sampled or how ties during bursting are resolved. For full details, please refer to the available Matlab source code (cf. section 1) that enables to recreate our results. The following section explains the application and adaptation of this algorithm for mobile robot localization.

4 Using HTM's Higher Order Sequence Memory for Mobile Robot Localization

4.1 Overview

Fig. 1 illustrates how HTM's higher order sequence memory is used for place recognition. Let us think of a robot that explores a new environment using a camera. It starts with an empty database and iteratively processes new image data while moving through the world. For each frame (or each n-th frame) it has to decide, whether the currently perceived scene is already in the database or not. This poses a set of binary decision problems, one for each image pair. The similarity matrix on the right side of Fig. 1 illustrates the possible outcome: each entry is the similarity of a current query image to a database image. To obtain binary decisions, a threshold on the similarity can be used. If we think of a continuously moving robot, it is useful to include information of previous frames to create these similarity values (cf. section 2 on sequence-based localization).

On an abstract level, the state of the cells in SHOSM (variable M in algorithm 1) is an encoding for the current input data in the context of previous observations. In terms of mobile robot localization, it provides an encoding of the currently observed place in the context of the prior trajectory to reach this place. All that remains to be done to use SHOSM for this task is to provide input and output interfaces. SHOSM requires the input to be encoded as sparse distributed representations. For example, we can think of a holistic encoding of the current camera image. More sophisticated encodings could also include local features and their relative arrangement similar to recent developments of HTM theory [12]. For several datatypes there are SDR encoders available [24]. Currently, for complex data like images and point clouds, there are no established SDR encoders, but there are several promising directions, e.g. descriptors based on sparse coding or sparsified descriptors from Convolutional Neural Networks [20]. Moreover, established binary descriptors like BRIEF or BRISK can presumably be sparsified using HTM's spatial pooler algorithm [9].

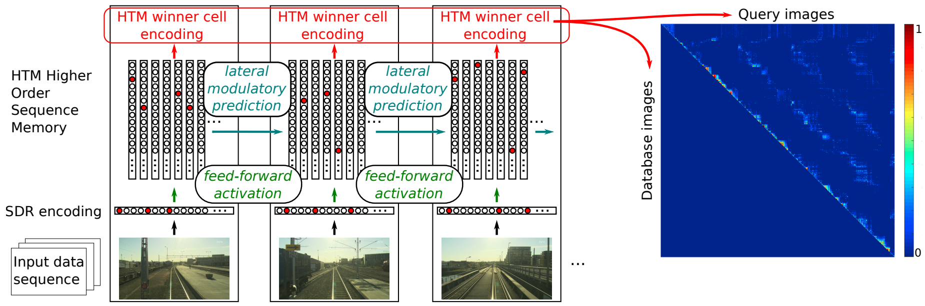

Fig. 1. Place recognition based on SHOSM winner cells. *(left)* Each frame of the input data sequence is encoded in form of a SDR and provides feed-forward input to the minicolumns. Between subsequent frames, active cells predict the activation of cells in the next time step. Output representation is the set of winner cells. *(right)* Example similarity matrix for a place recognition experiment with 4 loops (visible as (minor) diagonals with high similarity). The similarities are obtained from SDR overlap of the sparse vector of winner cells.

Output of SHOSM are the states of the cells, in particular a set of current winner cells. This is a high dimensional, sparse, binary code and the decision about place associations can be based on the similarity of these codes (e.g. using overlap of 1-bits [3]). If an input SDR activates existing minicolumns, this corresponds to observing an already known feature. If we also expected to see this feature (i.e. there are predicted cells in the active minicolumn), then this is evidence for revisiting a known place. The activation of the predicted cells yields a similar output code as at the previous visits of this place - this results in a high value in the similarity matrix. If there are no predicted cells, this is evidence for observation of a known feature at a novel place - thus unused (or rarely used) cells in these minicolumns become winner cells (cf. line 11 in algorithm 1). If there is no active minicolumn, we observe an unseen feature and store this feature in the database by creating a new set of minicolumns.

Using these winner-cell codes instead of the input SDRs directly, incorporates sequence information in the binary decision process. Experimental evidence for the benefit of this information will be provided in section 5.

4.2 Theoretical analogies of HTM and place recognition

This section discusses interesting theoretical association of aspects of HTM theory and the problem of mobile robot localization.

1. Minicolumns ⇔ Feature detectors Feature detectors extract distinctive properties of a place that can be used to recognize this place. In case of visual localization, this can be, for instance, a holistic CNN descriptor or a set of SIFT keypoints. In HTM, the sensor data is encoded in SDRs. Minicolumns are activated if there is a high overlap between the input SDR and the sampled connections of this minicolumn. The activation of a minicolumn corresponds to detecting a certain pattern in the input SDR - similar to detecting a certain CNN or SIFT descriptor.

2. Cells ⇔ Places with a particular feature The different cells in an active minicolumn represent places in the world that show this feature. All cells in a minicolumn are potentially activated by the same current SDR input, but in different context. In the above example of input SDR encodings of holistic image descriptors, the context is the sequence of encodings of previously seen images. In the example of local features and iteratively attending to individual features, the context is the sequence of local features.

3. Minicolumn sets ⇔ Ensemble classifier The combination of information from multiple minicolumns shares similarities to ensemble classifiers. Each minicolumn perceives different information of the input SDR (since they are not fully connected but sample connections) and has an individual set of predictive lateral connections. The resulting set of winner cells combines information from all minicolumns. If the overlap metric (essentially a binary dot product) is used to evaluate this sparse result vector, this corresponds to collecting votes from all winner cells. In particular, minicolumn ensembles share some properties of bagging classifiers [4] which, for instance, can average the outcome of multiple weak classifiers. However, unlike bagging, minicolumn ensembles do not create subsets of the training data with resampling, but use subsets of the input dimensions.

4. Context segments ⇔ Paths to a place Different context segments correspond to different paths to the same place. In the neurophysiological model, there are multiple lateral context segments for each cell. Each segment represents a certain context that preceded the activation of this cell. Since each place in the database is represented by a set of cells in different minicolumns, the different segments correspond to different paths to this place. If one of the segments is active, the corresponding cell becomes predicted.

5. Feed-forward segments ⇔ Different appearances of a place Although it is not supported by the neurophysiological model, there is another interesting association: If there were multiple feed-forward segments, they could be used to represent different appearances of the same place. Each feed-forward segment could respond to a certain appearance of the place and the knowledge about context of this place would be shared across all appearances. This is not implemented in the current system.

4.3 rSHOSM: SHOSM with additional randomized connections

Beyond the simplification of the higher order sequence memory described in section 3.2 we propose another beneficial modification of the original algorithm. The original SHOSM algorithm is designed to provide an individual representation of each element of a sequence dependent on its context. If anything in the context is changed, the representation also changes completely.

Fig. 2 illustrates this on a toy grid world with places A-F. What happens if a robot follows the red loopy trajectory $ABCDEBC$? At the first visit of place B, a representation is created that encodes B in the context of the previous observation A, lets write this as B_A. This encoding corresponds to a set of winner

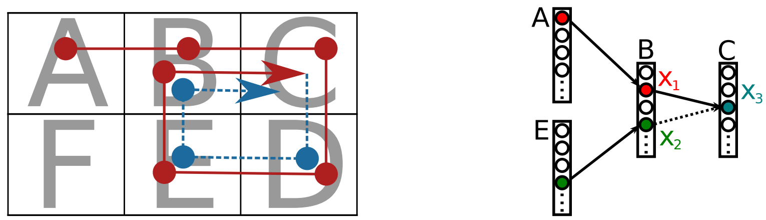

Fig. 2. *(left)* Toy example that motivates rSHOSM. See text for details. *(right)* Illustration of the loss of sequence information in case of multiple lateral connections from different cells x_1, x_2 of one minicolumn representing place B to a cell x_3. If the dotted connection from x_2 to x_3 exists, we can not distinguish the sequences (A,B,C) and (E,B,C) from an activation of x_3. Please keep in mind that in the actual system many parallel active minicolumns contribute to the representation of elements and sequences; for simplification, only a single minicolumn per element is shown.

cells. At the second visit of place B, there is a different context: the whole previous sequence $ABCDE$, resulting in an encoding B_{ABCDE}. The encodings B_A and B_{ABCDE} share the same set of active minicolumns (those that represent the appearance of place B) but completely different winner cells (since they encode the context). Thus, place B can not be recognized based on winner cells.

Interestingly, the encodings of C_{AB} and C_{ABCDEB} are identical. This is due to the effect of bursting: Since B is not predicted after the sequence $ABCDE$, all cells in minicolumns that correspond to B activate their predictions, including those who predict C (line 10 in algorithm 1). Thus, the place recognition problem appears only for the first place of such a loopy sequence. Unfortunately, this situation becomes worse if we revisit places multiple times, which is typical for a robot operating over a longer period of time in the same environment. The creation of unwanted unique representations for the same place affects one additional place each iteration through the sequence. For example, if the robot extends its trajectory to the blue path in Fig. 2, there will be a unique (not-recognizable) representation for places B and C at this third revisit. At a fourth revisit, there will be unique representations for B, C and D and so on.

Algorithmically, this is the result from a restriction on the learning of connections in line 14 of Algorithm 1: If the previously active minicolumn already has a connection to the currently active cell, then no new connection is created. Fig. 2 illustrates the situation. This behavior is necessary to avoid that two cells x_1, x_2 of a minicolumn predict the same cell x_3 in another minicolumn. If this would happen, the context (i.e., the sequence history) of the cell x_3 could not be distinguished between the contexts from cells x_1 and x_2.

To increase the recognition capabilities in such repeated revisits, we propose to alleviate the restriction on the learning of connections in line 14 of Algorithm 1: Since the proposed systems evaluates place matchings based on an ensemble decision (spread over all minicolumns), we propose to except the learning restriction for a small portion of lateral connections by chance. This is, to allow the creation of an additional new connection from a minicolumn to a cell, e.g., with a 5 % probability (i.e., to add the dotted connection from cell x_2 to x_3

in Fig. 2). Thus, some of the cells that contribute to the representation of a sequence element, do not provide a unique context but unify different possible contexts. This increases the similarity of altered sequences at the cost of reducing the amount of contained context. Since creating this connection once, introduces ambiguity for *all* previous context information for this cell, the probability of creating the additional connection should be low. This slightly modified version of the simplified higher order sequence memory is coined rSHOSM. The difference between SHOSM and rSHOSM is experimentally evaluated in the next section.

5 Experimental results

In this section, we demonstrate the benefit of the additional randomized connections from the previous section 4.3 and compare the presented approach against a baseline algorithm in a set of simulated place recognition experiments. We simulate a traversal through a 2D environment. The robot is equipped with a sensor that provides a 2,048 dimensional SDR for each place in the world; different places are grid-like arranged in the world. Using such a simulated sensor, we circumvent the encoding of typical sensor data (e.g. images or laser scans) and can directly influence the distinctiveness of sensor measurements (place-aliasing: different places share the same SDR) and the amount of noise in each individual measurement (repeated observations of the same place result in somewhat different measurements). Moreover, the simulation provides perfect ground-truth information about place matchings for evaluation using precision-recall curves: Given the overlap of winner cell encodings between all pairings in the trajectory (the similarity matrix of Fig.1), a set of thresholds is used, each splitting the pairings into matchings and non-matchings. Using the ground-truth information, precision and recall are computed. Each threshold results in one point on the precision-recall curves. For details on this methodology, please refer to [21].

Parameters are set as follows: input SDR size is 2,048; # 1-Bits in input SDR is 40; #cells per minicolumn is 32; #new minicolumns (Alg. 1, line 3) is 10; connectivity rate input SDR - minicolumn is 50%; and threshold on SDR overlap for active minicolumns is 25%.

5.1 Evaluation of additional randomized connections in rSHOSM

To demonstrate the benefit of the additional randomized connections in rSHOSM, we simulate a robot trajectory with 10 loops (each place in the loop is visited 10 times), resulting in a total of 200 observations. In this experiment, there are neither measurement noise nor place-aliasing in the simulated environment. The result can be seen on the left side of Fig. 3. Without the additional randomized connections, recall is reduced since previously seen places get new representations dependent on their context (cf. section 4.3).

5.2 Place recognition performance

This section shows results demonstrating the beneficial properties of the presented neurally inspired place recognition approach: increased robustness to place-aliasing

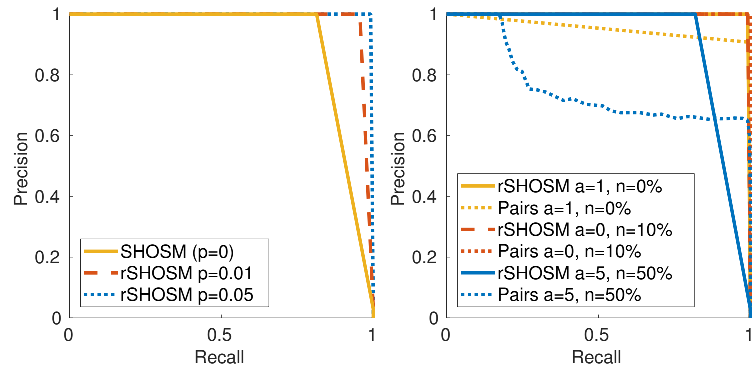

Fig. 3. *(left)* Benefit of the randomized connections in rSHOSM (with probabilities 0.01 and 0.05 of additional connections). This experiment does not involve noise or place-aliasing. *(right)* Comparison of the proposed rSHOSM with a baseline pairwise comparison in three differently challenging experiments. Parameter a is the amount of aliasing (the number of pairs of places with the same SDR representation) and n is the amount of observation noise (percentage of moved 1-bits in the SDR). In both plots, top-right is better.

and observation noise. Therefore, we compare the results to a simple baseline approach: brute-force pairwise comparison of the input SDR encodings provided by the simulated sensor. The right side of Fig. 3 shows the resulting curves for three experimental setups (each shown in a different color). We use the same trajectory as in the previous section but vary the amount of observation noise and place-aliasing. The noise parameter n controls the ratio of 1-bits that are erroneously moved in the observed SDR. For instance, $n = 50\%$ indicates that 20 of the 40 1-bits in the 2,048 dimensional input vector are moved to a random position. Thus, only 20 of the 2,048 dimensions can contribute to the overlap metric to activate minicolumns. The place-aliasing parameter a counts the number of pairs of places in the world which look exactly the same (except for measurement noise). For instance, $a = 5$ indicates that there are 5 pairs of such places and each of these places is visited 10 times in our 10-loops trajectory.

Without noise and place-aliasing, the baseline approach provides perfect results (not shown). In case of measurement noise (red curves), both approaches are amost not effected, due to the noise robustness of SDRs. In case of place-aliasing (yellow curves), the pairwise comparison can not distinguish the equivalently appearing places resulting in reduced precision. In these two experiments with small disturbances, the presented rSHOSM approach is not affected. The blue curves show the results from a challenging combination of high place-aliasing and severe observation noise - a combination that is expected in challenging real world place recognition tasks. Both algorithms are affected, but rSHOSM benefits from the usage of sequential information and performs significantly better than the baseline pairwise comparison.

In the above experiments, typical processing time of our non-optimized Matlab implementation of rSHOSM for one observation is about 8 ms using a standard laptop with an i7-7500U CPU @ 2.70GHz.

6 Discussion and Conclusion

The previous sections discussed the usage of HTM's higher order sequence memory for visual place recognition, described the algorithmic implementation and motivated the system with a discussion of theoretical properties and some experimental results where the proposed approach outperformed a baseline place recognition algorithm. However, all experiments used simulated data. The performance on real world data still has to be evaluated. Presumably, the presented benefit above the baseline could also be achieved with other existing techniques (e.g. SeqSLAM). It will be interesting to see, whether the neurally inspired approach can address some of the shortcomings of these alternative approaches (cf. section 2). Such an experimental comparison to other existing place recognition techniques should also include a more in-depth evaluation of the parameter of the presented system. For the presented initial experiments, no parameter optimization was involved. We used default parameters from HTM literature (which in turn are motivated by neurophysiological findings).

The application on real data poses the problem of suitable SDR encoders for typical robot sensors like cameras and laser scanners - an important direction for future work. Based on our previous experience with visual feature detectors and descriptors [20,21,19], we think this is also as a chance to design and learn novel descriptors that exploit the beneficial properties of sparse distributed representations (SDRs). An interesting direction for future work would also be to incorporate recent developments on HTM theory on processing of local features with additional location information - similar in spirit to image keypoints (e.g. SIFT) that are established for various mobile robot navigation tasks.

Although, the presented place recognition approach is inspired by a theory of the neocortex, we do not claim that place recognition in human brains actually uses the presented algorithm. There is plenty of evidence [8] of structures like entorhinal grid cells, place cells, head direction cells, speed cells and so on, that are involved in mammal navigation and are not regarded in this work.

The algorithm itself also has potential theoretical limitations that require further investigation. For example, one simplification from the original HTM higher order sequence memory is the creation of new minicolumns for unseen observation instead of using a fixed set of minicolumns. This allows a simple one-shot learning of associations between places. In a practical system the maximum number of minicolumns should be limited. Presumably, something like the Hebbian-like learning in the original system could be used to resemble existing minicolumns. It would be interesting to evaluate the performance of the system closer to the capacity limit of the representation.

Finally, SDRs provide interesting theoretical regarding runtime and energy efficiency. However, this requires massively parallel implementations on special hardware. Although this is far beyond the scope of this paper, in the future, this might become a unique selling point for deployment of these algorithms on real robots.

References

1. Numenta. `https://numenta.com/`, accessed: 2018-05-09
2. Nupic. `https://github.com/numenta/nupic`, accessed: 2018-05-09
3. Ahmad, S., Hawkins, J.: Properties of sparse distributed representations and their application to hierarchical temporal memory. CoRR **abs/1503.07469** (2015)
4. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (Aug 1996). https://doi.org/10.1023/A:1018054314350
5. Cadena, C., Galvez-Lpez, D., Tardos, J.D., Neira, J.: Robust place recognition with stereo sequences. IEEE Transactions on Robotics **28**(4), 871–885 (Aug 2012). https://doi.org/10.1109/TRO.2012.2189497
6. Cui, Y., Ahmad, S., Hawkins, J.: Continuous online sequence learning with an unsupervised neural network model. Neural Computation **28**(11), 2474–2504 (nov 2016). https://doi.org/10.1162/NECO_a_00893
7. Engelson, S., McDermott, D.: Error correction in mobile robot map-learning. In: Int. Conf. on Robotics and Automation (ICRA). pp. 2555–2560 (1992)
8. Grieves, R., Jeffery, K.: The representation of space in the brain. Behavioural Processes (12 2016)
9. Hawkins, J., Ahmad, S., Purdy, S., Lavin, A.: Biological and machine intelligence (bami) (2016), `https://numenta.com/resources/biological-and-machine-intelligence/`, initial online release 0.4
10. Hawkins, J.: On Intelligence (with Sandra Blakeslee). Times Books (2004)
11. Hawkins, J., Ahmad, S.: Why neurons have thousands of synapses, a theory of sequence memory in neocortex. Frontiers in Neural Circuits **10**, 23 (2016). https://doi.org/10.3389/fncir.2016.00023, `https://www.frontiersin.org/article/10.3389/fncir.2016.00023`
12. Hawkins, J., Ahmad, S., Cui, Y.: A theory of how columns in the neocortex enable learning the structure of the world. Frontiers in Neural Circuits **11**, 81 (2017). https://doi.org/10.3389/fncir.2017.00081
13. Ho, K.L., Newman, P.: Detecting loop closure with scene sequences. International Journal of Computer Vision **74**(3), 261–286 (Sep 2007). https://doi.org/10.1007/s11263-006-0020-1
14. Johns, E., Yang, G.: Dynamic scene models for incremental, long-term, appearance-based localisation. In: Proc. of International Conference on Robotics and Automation (ICRA). pp. 2731–2736. IEEE (2013)
15. Lowry, S., Sunderhauf, N., Newman, P., Leonard, J.J., Cox, D., Corke, P., Milford, M.J.: Visual place recognition: A survey. Trans. Rob. **32**(1), 1–19 (Feb 2016). https://doi.org/10.1109/TRO.2015.2496823
16. Milford, M., Wyeth, G., Prasser, D.: Ratslam: a hippocampal model for simultaneous localization and mapping. In: Proc. of International Conference on Robotics and Automation (ICRA). pp. 403–408. IEEE (2004)
17. Milford, M., Wyeth, G.F.: Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In: Proc. of International Conference on Robotics and Automation (ICRA). pp. 1643–1649. IEEE (2012)
18. Naseer, T., Spinello, L., Burgard, W., Stachniss, C.: Robust visual robot localization across seasons using network flows. In: Proc. of. AAAI Conference on Artificial Intelligence. pp. 2564–2570. AAAI'14, AAAI Press (2014)
19. Neubert, P.: Superpixels and their Application for Visual Place Recognition in Changing Environments. Ph.D. thesis, Chemnitz University of Technology (2015), `http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-190241`

20. Neubert, P., Protzel, P.: Local region detector + CNN based landmarks for practical place recognition in changing environments. In: Proc. of European Conference on Mobile Robotics (ECMR). pp. 1–6. IEEE (2015)
21. Neubert, P., Protzel, P.: Beyond holistic descriptors, keypoints, and fixed patches: Multiscale superpixel grids for place recognition in changing environments. IEEE Robotics and Automation Letters **1**(1), 484–491 (2016)
22. Neubert, P., Schubert, S., Protzel, P.: Learning vector symbolic architectures for reactive robot behaviours. In: Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS) Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics (2016)
23. Nilsson, N.J.: Shakey the robot. Tech. Rep. 323, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025 (Apr 1984)
24. Purdy, S.: Encoding data for htm systems. CoRR **abs/1602.05925** (2016)
25. Spruston, N.: Spruston n. pyramidal neurons: dendritic structure and synaptic integration. nat rev neurosci 9: 206-221 **9**, 206–21 (04 2008)