Sparse and topological coding for visual localization of autonomous vehicles

Sylvain Colomer, Nicolas Cuperlier, Guillaume Bresson, Steve Pechberti, Olivier Romain

Sparse and topological coding for visual localization of autonomous vehicles

Sylvain Colomer[1,2], Nicolas Cuperlier[1], Guillaume Bresson[2], Steve Pechberti[2], and Olivier Romain[1]

[1] Laboratoire ETIS UMR8051, Université Paris Seine, ENSEA, CNRS, France.
https://www.etis-lab.fr/
[2] Institut VEDECOM, 23 bis Allée des Marronniers, 78000 Versailles, France.
https://www.vedecom.fr/

Abstract. Efficient encoding of visual information is essential to the success of vision-based navigation tasks in large-scale environments. To do so, we propose in this article the Sparse Max-Pi neural network (SMP), a novel compute-efficient model of visual localization based on sparse and topological encoding of visual information. Inspired by the spatial cognition of mammals, the model uses a "topologic sparse dictionary" to efficiently compress the visual information of a landmark, allowing rich visual information to be represented with very small codes. This descriptor, inspired by the neurons in the primary visual cortex (V1), are learned using sparse coding, homeostasis and self-organising map mechanisms. Evaluated in cross-validation on the Oxford-car dataset, our experimental results show that the SMP model is competitive with the state of the art. It thus provides comparable or better performance than CoHog and NetVlad, two state-of-the-art VPR models.

Keywords: visual place recognition · sparse coding · autonomous vehicle · visual cortex · bio-inspired robotics.

1 Introduction

The problem of autonomous navigation on a robotised vehicle requires the simultaneous resolution of a multitude of issues [11]. Indeed, driving a vehicle on an open road requires a lot of techniques and knowledge, even to carry out simple navigation tasks in favorable environments. To name a few, a vehicle must be at the same time able to locate accurately its position, stay in its lane, avoid accidents and respect traffic rules. Consequently, the models of autonomous vehicle are often based on the use of numerous modules, specialized in one or more sub-problems.

Among the different modules traditionally used in navigation, the localization one is particularly important for its central role in the navigation task. Its performance can severely limit the ability of a navigation system to complete a navigation task. To reach the best performance, the module is usually made with powerful sensors such as GNSS or LiDAR. Although highly efficient, these sensors suffer from high cost and significant operating limitations [11]. This leads

to the development of new techniques relying on different modalities, notably the methods of Visual place recognition (VPR) that propose to use visual information as a source of localization, since cameras are rich, inexpensive and low consumption sensors.

In recent years, a lot of VPR models have been proposed in various application fields such as robotics, big data or machine vision. However, despite significant progress in terms of performance and computing time, the methods proposed are still struggling to provide a complete alternative for the localization system of autonomous cars. Although visual space is a very rich source of information, this space is also the object of a strong dynamic which makes its use more complicated, especially when moving to large scales of time and distance. To operate even on a single day, the methods proposed must be robust at the same time to multiple issues like lighting problems, changing weather or variation in human activity. However, obtaining such performance is often accompanied by a high computational cost that limits the use of such a system to small scales of deployment. This implies finding a method for representing visual information that is light enough to limit its computational cost, while maintaining enough information to distinguish locations under varying visual conditions.

Up to now, the only system known which has the ability to build such a representation are biological ones. Indeed, several studies have shown that certain species of mammals are able to perform large-scale trajectories by relying essentially on the vision [4]. This ability is thought to rely on several key structures in mammals brain, notably the hippocampal system (HS) and the visual cortex (VC). HS would thus be involved in the memory processes of animal cognition and would contain a neuronal map of the environment [8]. On the other hand, VC would be responsible for the encoding the visual information upstream to the HS. Thus, several studies suggest that the first layers of the visual cortex use mechanisms similar to sparse coding methods for representing visual information. VC would break down the visual information into elementary patterns a bit like a visual sparse dictionary, except that it would respect a topological arrangement of patterns.

Following a bio-inspired approach, we present in this paper the following contributions:

- We propose the topologic sparse coding (TSC) algorithm, a new method of sparse coding intended for the encoding of visual information for localization. This method allows the construction of a topologic sparse dictionary, i.e. a dictionary of visual features that respects both a constraint of sparsity and of spatial topology. The topology allows, unlike a classical sparse dictionary, to build a structured dictionary where neurons coding for similar features are physically close in the dictionary. As a result, similar images have much closer codes and are therefore much easier to recognise.
- We propose the Sparse Max-Pi (SMP) neural network, a novel model of VPR for autonomous vehicles. This model, based by the LPMP model, allows to build in an unsupervised way a neural representation of the environment. Unlike the original model, the SMP model uses a topological sparse dictionary

to encode information. By this way, the system uses a more compact code to represent landmarks, allowing to strongly divide the memory cost of a place while maintaining equivalent (or slightly better) localization performance.
– We evaluated the SMP model on the OxfordCar dataset in cross-validation and compare its performances with two lead models of VPR : CoHog [12] and NetVLAD [1].

The remaining of this paper is organized as follows: On first, we describe the TSC algorithm and the SMP model. Afterwards, we present the experiments carried out and their results. The last part is left to the conclusions.

2 Related work : definition of the sparse coding

Sparse Coding (SC) refers to the set of methods that aim to construct a "sparse" representation for a data space, i.e. a representation that can be characterised by its propensity to encode data through the activity of a small number of its components. This representation, called dictionary, has the advantage of efficiently capturing redundant patterns in the data space, allowing to build an efficient code for the data space. To generate a sparse representation, a "sparsity criteria" is defined, whose purpose is to determine towards which definition of sparsity the system should converge. The most common is to use the l0 norm of the sparse code (the code generate by the representation), as defined by the following equation:

$$\boldsymbol{s} = \sum_{m=1}^{M} a_m \boldsymbol{\phi}_m \quad \text{subject to} \quad \min_{\boldsymbol{a}} \|\boldsymbol{a}\|_0 \tag{1}$$

where $s \in \mathbb{R}^n$ is the input vector, a_m is an coefficient of activity, $\|.\|_0$ the L_0 norm and $\phi_m \in \mathbb{R}^N$ is an element of the dictionary $\boldsymbol{\Phi}$ called an atom. In this equation, the L_0 norm imposes that the system must converge to a representation using as many atoms as possible to represent a data while retaining its ability to reconstruct all the patterns in the data space. Contrary to other methods such as a PCA, sparse coding builds an overcomplete base of a data space. It allows to better capture the general patterns of the space and to achieve a more efficient representation of the input space.

3 Topological sparse coding

The topological sparse coding (TSC) algorithm is a new method of SC intended for the encoding of visual information. Based on the Sparse Hebbian Learning algorithm [7], this method allows to learn from a set of data a "topological sparse dictionary" i.e. a sparse dictionary arranged on a 2D grid like a Kohonen network. Thus, unlike a traditional sparse coding algorithm, the dictionary has two dimensions and respects a topology. Consequently, neurons coding for similar features are physically close in the dictionary, allowing to have similar codes for two close images, which eases their comparison.

TSC was first designed to compress visual information in the same way as the "primary visual cortex", known to be the first stage of visual information encoding in mammals. In particular, it reproduces the functioning and the organisation of the first layer of neurons, composed of "single cells". These neurons, sensitive to specific orientations, are used by the visual cortex to break down visual information into robust elementary patterns [9]. Thus, the use of a sparse coding mechanism enables the construction of neurons with receiver fields similar to simple cells [6]. The addition of the topology allows to reproduce the retinotopic structure of the visual cortex, where neurons sensitive to similar orientations are side by side.

To build a dictionary of size (M, N), TSC uses a large batch of images (in our case, thumbnails of landmarks), pre-processed with a whitening filter [7]. It produces a sparse dictionary by alternating between two processes: an *encoding stage* where the current image is reconstructed with a limited number of atoms, and an *update stage* where the dictionary is modified to improve it reconstruction performance, depending of the encoding result.

Encoding stage: To encode an image, TSC uses an homeostatic version of the "Matching Pursuit" algorithm. The objective of this algorithm is to recursively find the best combination of atoms and activity in the dictionary that allows a better reconstruction of the input signal. The number of atoms for the reconstruction of the signal is limited by the value N_0, which indirectly controls the level of sparsity of the dictionary. Thus, at an iteration t, the system searches in the dictionary for the best pair of atom/activity that minimizes the intermediate reconstruction error $r(t)$, such as :

$$r(t) = I - \sum_{n=1}^{N} \sum_{m=1}^{M} \hat{a}_{m,n}(t)\phi_{m,n} \tag{2}$$

where $\hat{a}_{m,n}$ is the current activity vector. Carried out N_0 times, this mechanism allow to find a set of atoms that reconstructs the signal, given the current dictionary. To prevent an atom from becoming dominant in the reconstruction process, the value of $\hat{a}_{m,n}(t)$ is transformed by $z(.)$, a homeostasis function. Thus, $z(.)$ modifies the activity of the sparse code according to its activity distribution to balance the atoms that are over or under activated. It can be described by the next equations:

$$s^* = \text{ArgMax}_{m,n}\left[z_{m,n}\left(\hat{a}_{m,n}\right)\right] \tag{3}$$

$$\text{with} \quad z_{m,n}\left(\hat{a}_{m,n}\right) \leftarrow \left(1 - \eta_h\right) z_{m,n}\left(\hat{a}_{m,n}\right) + \eta_h P\left(a_{m,n} \leq \hat{a}_{m,n}\right) \tag{4}$$

Where s^* is the index of the selected atom and η_h is the homeostatic coefficient.

Update of dictionary: The update of the dictionary is realized with an an Hebb's rule, formulated for an atom $\phi_{m,n}$ as:

$$\begin{cases} \phi_{m,n} \leftarrow \phi_{m,n} + \eta * (\mathbf{I} - \phi\mathbf{a}) * a_{m,n} & \text{if } a_{m,n} > 0 \\ \phi_{m,n} \leftarrow \phi_{m,n} + \eta * (\mathbf{I} - \phi\mathbf{a}) * h(r, \mathbf{s}) & \text{otherwise} \end{cases} \tag{5}$$

where η is the learning rate, $\mathbf{s}$ is the list of atoms that were selected during the encoding, r is the current atom position and $h(.)$ the function that defines the neighbourhood membership of an atom. $h(.)$ computes the smallest distance between the current atom's position and each atom selected in the encoding process, as described in the next equation:

$$h(r, \mathbf{s}) = \exp\left(-\frac{Min_{s \in \mathbf{s}}(||l - s||)}{2\sigma^2(t)}\right) \tag{6}$$

with $||.||$ a distance function and σ the neighbourhood coefficient. Thus, a neuron is updated in two cases : if it has contributed to the reconstruction of the signal ($a_{m,n} > 0$) or if it is in the neighbourhood of one of them.

4 SMP model

The Sparse Max-Pi (SMP) model is an unsupervised neural architecture aimed at solving the problem of visual localization for autonomous vehicle. It uses the vehicle's visual information and absolute orientation[3] to builds a neural representation of an environment. In particular, the model simulates place cells, a specific king of neurons found in the HS of mammals. Like biological ones, these neurons respond with high activity for a given place in the environment and have shown interesting properties of robustness in complex environments [2]. The SMP model was initially proposed to resolve the computational cost issues of the LPMP model on which it is strongly based. Unlike the original model, it does not use a log polar to encode visual information but a topological sparse dictionary. This allows the SMP model to use much smaller visual descriptors, strongly reducing its computational cost[4].

To locate an image, the SMP model starts by searching for the position of its N_p most significant landmarks. To do so, a saliency map is built by successively convolving the image with two filters (a Deriche filter and a Difference of Gaussians (DoG) filter) to highlight its curvature points. The landmarks are then sorted and selected via a local competition, a bio-inspired mechanism which allows to keep only the most significant landmark. Then, two information streams are processed in parallel for each detected landmark:

1. The *visual identity* (or *what pathway*), corresponding to the encoding of a local view centered on the position of the landmark. To compress the visual information, the system uses a topological sparse dictionary followed by a max-pooling layer of 2x2.
2. The *azimuth information* (or *where pathway*), corresponding to the encoding of the absolute direction of the landmark in the environment. This information is computed using the vehicle absolute orientation and the PoI coordinate in the image.

[3] Absolute orientation can be obtained from a magnetic or visual compass [2]

[4] The model must compare the current image with all images stored in memory to localize a place. The smaller the code, the faster the memory search.

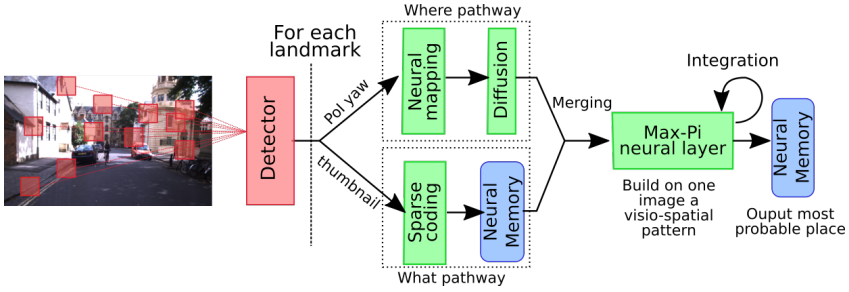


Fig. 1. Scheme of the SMP model. To locate a place, the SMP model encodes the visual identity and absolute orientation of each landmark in an image, and merges them into a Max-Pi layer to build a visuo-spatial model. These two information are encoded for each landmark through two pathways: the "what pathway", where a sparse dictionary is used to encodes the landmarks and store them in a memory; And the "where pathway", where the model encodes the absolute orientation of the landmarks in arrays of neurons.

Finally, the visual identity and the azimuth information of each landmark are merged and accumulated in a Max-Pi neural layer. This structure allows to build a visuo-spatial code which is characteristic of the vehicle current position (see [2] for a more complete description). This visuo-spatial code is then sent to a neural memory called Winner Memory, to be either memorized or searched through the known locations.

5 Materials and methods

5.1 Dataset

Among the different datasets available, we decided to use the *Oxford-car dataset* [5] to evaluate the performance of the SMP model. This dataset has the advantage of being very complete, giving access to a hundred driving sequences on the same road in various conditions (lighting, traffic, or weather). As each sequence is 9 kilometers long, we decided to subdivide the dataset into several test sequences. To do so, we divided the dataset into different routes through different environments (*city-center*, *boulevard* and $forest$), as proposed in previous work [2]. For each route, 3 different trajectories have been selected, taken at different times. The sequences were selected to present favorable weather conditions, but different levels of human activity. Moreover, the different dictionaries tested were learned from a dataset of 54000 landmarks, generated using the visual system of SMP. They are extracted from the sequence "2014/07/14 14:49:50", outside the areas of performance evaluation.

5.2 Metrics

To evaluate the TSC algorithm and the SMP model, we selected four metrics :

Env.	Images	Distance (meters)	Duration (seconds)	Sequence date	Reference index
Boulevard	1401	625	89	2014/07/14 14:49:50	2820-4220
Boulevard	1159	624	74	2015/07/29 13:09:26	5928-7086
Boulevard	1572	626	101	2015/08/4 14:54:57	5665-7236
City-center	1521	532	104	2015/05/19 14:06:38	6199-7719
City-center	2227	527	143	2015/07/29 13:09:26	7210-9436
City-center	2134	585	140	2015/05/22 11:14:30	7728-9861
Forest	927	292	61	2014/07/14 14:49:50	5190-6116
Forest	566	286	38	2015/05/19 14:06:38	7827-8392
Forest	595	287	37	2015/05/22 11:14:30	10211-10805

Table 1. Trajectories selected from the *Oxford-car dataset.* This table presents the trajectories selected in the dataset to assess the performances of the SMP model. Three different environments were selected to vary the localization conditions.

- **Reconstruction error:** The reconstruction error of a dictionary is the difference between a data pattern and its reconstructed image after encoding.
- **Population kurtosis:** This metric measures the distribution of dictionary responses to a single stimulus. It has been described as the best metric for measuring the sparsity of a dictionary [10].
- **Precision/recall**: The localization performances were evaluated using standard precision/recall measurements. To do so, we followed the classical method for evaluating VPR systems, which consists of characterising the distance between the coordinate of an image to localize and the coordinates of the image that the model best recognises [2]. These results are summarized by their Area Under Curves (AUC) and the recall at 100% precision.
- **Response frequency**: It was assessed by measuring the average frequency that each model takes to answer a query, depending of the number of locations learned.

5.3 Evaluation methodology

To study the SMP model, three types of experiments were performed:

- **Learning experiments:** The first type of experiment is performed to study the evolution of the code generated by the SMP model during training. For this purpose, the training of the SMP model is interrupted at regular intervals in order to test it's dictionary on a subset of the training dataset. This subset is then encoded via its sparse dictionary and evaluated using the population Kurtosis and the reconstruction error.
- **Localization performance experiments:** The second type of experiment allows the study of the localization performance of the VPR model under different conditions. It is measured using standard precision/recall measurements in cross validation. Thus, for a given configuration (learning sampling rate or dictionary size), 3 tests are performed to address every configuration of learning and test sequence. Moreover, for the test sequences, one image

per meter is tested to see to what extent a vehicle using this model would be able to locate itself along all the path.
- **Computational cost experiments:** This experiment allows the study to measure the computational cost of a model in different conditions. To do so, we evaluated the average time taken by a model to respond to a query. To limit the influence of code optimization on the experiment, the models were evaluated using only one CPU and no GPU.

5.4 Implementation details

The performance of the SMP model relies on the tuning of two important parts: the visual system which influences the quality of the PoI (stable position, attachment on characterising landmarks) and the encoding system which influences the amount of information kept by the model. In this paper, we have focused our analyses on the encoding system, being the major novelty brought to the model. Thus, excepted for the dictionary, the parameters of the SMP model are the same than those use in the LPMP model [2]. Moreover, to compare SMP with CoHog and NetVLAD, we used the original implementation provided by their authors. Thus, we used the best pre-trained model for NetVLAD (VGG-16+whitening+Pittsburgh). Furthermore, experiments on localization performance were realized using an AMD Ryzen Threadripper 2990wx (3.7GHz) and experiments on computational cost were carried out using an Intel Core i9-9880H (2.3Ghz).

6 Results

6.1 Properties of TSC during learning

The figure 2 shows the evolution of the mean reconstruction error and the mean population Kurtosis during the training of a 30*30 dictionary. Thus, the first graph shows that during learning, the dictionary improves its ability to reconstruct the landmarks that are presented to it. The second graph on the other hand shows an increase in the mean population Kurtosis during the learning. This indicates that the sparsity of the code generated by the TSD increases, despite the addition of a topology constraint. These results seem to indicate that the algorithm does converge towards a solution that better encodes visual information and that is more sparse.

6.2 Evaluation of configuration/performance

The figure 3 shows the evolution of the SMP model's localization performance, according to its configuration[5]. They were computed on the first 250 meters of the *Boulevard* dataset with a sampling rate of 5m. Thus, the two graphs show that :

[5] To facilitate the notation of the dictionary configuration used during an experiment, the SMP model using a dictionary of size $n * n$ is called $SMP - n$.

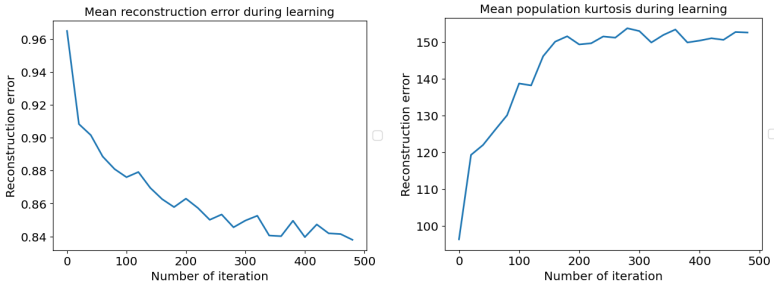


Fig. 2. Evolution of a topological sparse code during its learning process. Evolution of mean reconstruction error (left) and mean population kurtosis (right) at regular intervals during the learning process a 30*30 dictionary.

- **Increasing the number of atoms of the SMP model tends to improve its localization performance.** Moving from SMP-12 to SMP-30 improves the mean AUC by 8%. However, a peak of performance is reached at SMP-30.
- **SMP-30 has an average AUC equivalent to NetVlad.** SMP-30 has a mean AUC of 0.934, very close to that of NetVlad of 0.928.
- **SMP-30 has a slightly better average AUC than CoHog.** SMP-30 has an average AUC 5% better than CoHog.
- **All configurations of the SMP model have a better recall at 100% than the other models.** SMP-12, the least efficient configuration, has a recall at 100% of 0.136 against 0.063 for the CoHog model and 0.064 for the NetVlad model. This trend can also be observed in the figure 5 with a sampling of 5m.

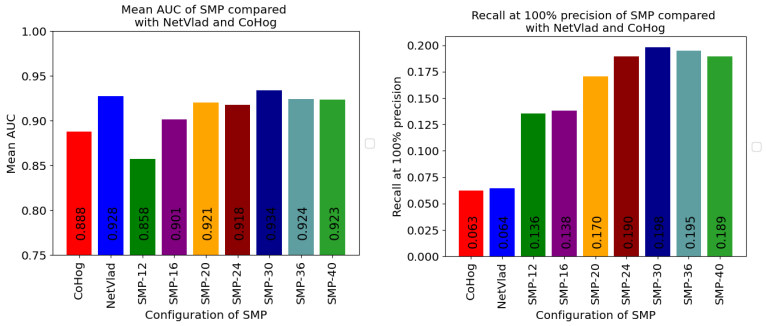
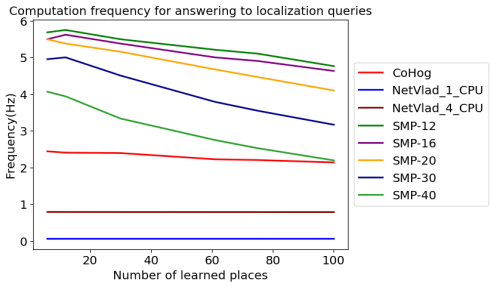


Fig. 3. Influence of SMP configuration on localization performances. The graphs presents the mean AUC (left) and the recall at 100% precision (right) of NetVlad, CoHog, and different configurations of the SMP model. The curves were computed in cross validation on the first 250 metres of the *Boulevard* dataset with a sampling rate of 5m.

To complete the previous results, the figure 4 shows the influence of the SMP configuration on the computational cost and the learning time. The graph on the left shows the average frequency at which the models answer to localization queries, according to the number of learned places. It allows us to conclude that:

- **Increasing the size of the SMP model decreases the computation frequency.** Going from SMP-20 to SMP-30 for 100 learned locations decreases the computation frequency from 4Hz to 3Hz.
- **At equivalent AUC, the SMP model has a higher average computing frequency than the NetVlad model.** SMP-30 allows to achieve a gain of $\times 60$ on the computation frequency with NetVlad-1-CPU[6] and a gain of $\times 3$ with NetVlad-4-CPU.
- **At equivalent AUC, the SMP model also has a higher average computation frequency than the CoHog model.** SMP-18 allows to achieve a gain of $\times 2$ on the computation frequency for 100 learned locations. However, unlike the SMP model and NetVlad, the CoHog model does not require learning to operate.



Model	Learning time	Descriptor size
SMP-12	215s	36
SMP-16	387s	64
SMP-20	538s	100
SMP-30	481s	225
SMP-36	1376s	324
SMP-40	1632s	400

Fig. 4. Evaluation of the computational cost of the SMP model The graph on the left presents the frequency at which SMP, CoHog and NetVLAD answer to localization queries depending on the number of locations learned. The table on the right shows the learning time for different configurations of the SMP model and the size of the descriptor used by the SMP model to encode a landmark.

The table on the right gives the learning time for each SMP configuration and the size of the feature vector used to represent a landmark. It shows that increasing the size of the dictionary has a strong impact on the learning time. In general, the larger the dictionary size, the longer the learning time. Thus, going from SMP-30 to SMP-40 multiplies the learning time by 5.

6.3 Evaluation of localization performances with the state of the art

The figure 5 shows the average performance SMP-30, CoHog and NetVlad according to the three environments and three sampling distances: 3m, 5m and 10m. The graphs on top show the mean AUC of the precision-recall curves and

[6] The NetVlad model run at an average frequency of 0.05 Hz with one CPU

the graphs below show the mean recall at 100%. Each value was computed in cross validation on the first 500 metres of each environment.

- **The SMP model has better localization performance than the CoHog model in almost all cases.** This difference is generally of 5% for the mean AUC, but is much larger for the recall at 100% precision. CoHog only exceeds SMP in the city-centre environment with a sampling distance of 2m.
- **The SMP model has competitive localization performance with the NetVlad model.** The SMP model has slightly better performance than the NetVlad model at a sampling distance of 2m and 5m but the NetVlad model is better at a sampling distance of 10m.

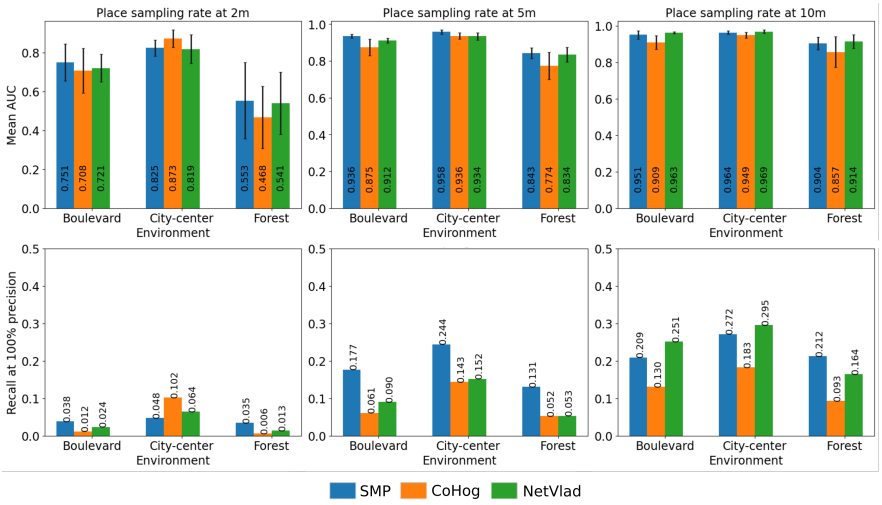


Fig. 5. Localization performances of SMP-30, CoHog and NetVLAD models. The performances are evaluated by computing precision/recall curves, summarized by their Area Under Curves and their recall at 100% precision. The evaluation was made in cross-validation on each environment and each place sampling rate.

7 Discussion and conclusion

In this paper, we proposed the TSC algorithm, a new method of sparse coding that allows to build an organized sparse dictionary. This method, contrary to the classic sparse coding method, allows to build a more "coherent" dictionary i.e with closer codes for two similar images. We have thus demonstrated that TSC have a strong interest in the context of visual localization to encode visual information. In particular, we demonstrated that the SMP model, based on the use of TSC, is competitive with two state-of-the-art models: CoHog and NetVlad.

Moreover, the experiments carried out demonstrated the interest of sparse coding for large-scale localization. In particular, we have seen that the model

allows to strongly compress the visual information while keeping very high localization performances. The performances obtained can thus allow us to hope for meeting the real-time constraints of a localisation system on an autonomous vehicle. Tests have been undertaken on real vehicles and have shown encouraging first results.

Finally, this proposal is in line with previous work, in particular the HSD model, an encoding model based on the use of several layers of sparse dictionaries [3]. Unlike the previous model, the TSC algorithm directly integrates the topology into the update rule. This writing of the algorithm results in a better sparse dictionaries, which perform more efficiently when used with a VPR model. Thus, further work has been undertaken to chain several TSC to further improve the performance of the model, in the same way as proposed in the HSD model.

References

1. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. arXiv:1511.07247 [cs] (May 2016)
2. Colomer, S., Cuperlier, N., Bresson, G., Gaussier, P., Romain, O.: Lpmp: A bio-inspired model for visual localization in challenging environments. Frontiers in Robotics and AI **8** (2022)
3. Colomer, S., Cuperlier, N., Bresson, G., Romain, O.: Forming a sparse representation for visual place recognition using a neurorobotic approach. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). pp. 3002–3009 (2021)
4. Geva-Sagiv, M., Las, L., Yovel, Y., Ulanovsky, N.: Spatial cognition in bats and rats: from sensory acquisition to multiscale maps and navigation. Nature Reviews Neuroscience **16**(2), 94–108 (Feb 2015). https://doi.org/10.1038/nrn3888
5. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 Year, 1000km: The Oxford RobotCar Dataset. The International Journal of Robotics Research (IJRR) **36**(1), 3–15 (2017)
6. Olshausen, B., Field, D.: Sparse coding of sensory inputs. Current Opinion in Neurobiology **14**(4), 481–487 (Aug 2004). https://doi.org/10.1016/j.conb.2004.07.007
7. Perrinet: An adaptive homeostatic algorithm for the unsupervised learning of visual features. Vision **3**(3), 47 (Sep 2019). https://doi.org/10.3390/vision3030047
8. Rolls, E.T., Wirth, S.: Spatial representations in the primate hippocampus, and their functions in memory and navigation. Progress in Neurobiology **171**, 90–113 (2018)
9. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(3), 411–426 (2007). https://doi.org/10.1109/TPAMI.2007.56
10. Willmore, B., Tolhurst, D.: Characterizing the sparseness of neural codes. Network: Computation in Neural Systems **12**(3), 255–270 (Jan 2001)
11. Yurtsever, E., Lambert, J., Carballo, A., Takeda, K.: A survey of autonomous driving: Common practices and emerging technologies. IEEE Access **8**, 58443–58469 (2020). https://doi.org/10.1109/ACCESS.2020.2983149
12. Zaffar, M., Ehsan, S., Milford, M., McDonald-Maier, K.: Cohog: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments. IEEE Robotics and Automation Letters **5**(2), 1835–1842 (Apr 2020)