Learning Visual Landmarks for Localization with Minimal Supervision

Muhammad Haris, Mathias Franzius, Ute Bauer-Wersing

2022

Preprint:

This is a post-peer-review, pre-copyedit version of an article published in International Conference on IMAGE ANALYSIS AND PROCESSING. The final authenticated version is available online at:
https://doi.org/10.1007/978-3-031-06427-2_64

Learning Visual Landmarks for Localization with Minimal Supervision

Muhammad Haris[1], Mathias Franzius[2], and Ute Bauer-Wersing[1]

[1] Frankfurt University of Applied Sciences, 60318 Frankfurt, Germany
[2] Honda Research Institute Europe GmbH, 63073 Offenbach, Germany

Abstract. Camera localization is one of the fundamental requirements for vision-based mobile robots, self-driving cars, and augmented reality applications. In this context, learning spatial representations relative to unique regions in a scene with Slow Feature Analysis (SFA) has demonstrated large-scale localization. However, it relies on hand-labeled data to train a CNN for recognizing unique regions. We propose a new approach that uses pre-trained CNN-detectable objects as anchors to label and learn new landmark objects or regions in a scene using minimal supervision. The method bootstraps the landmark learning process and removes the need to manually label large amounts of data. The anchor objects are only required to learn the new landmarks and become obsolete for the unsupervised mapping and localization phases. We present localization results with the learned landmarks in simulated and real-world outdoor environments and compare the results to SFA on complete images and PoseNet. The landmark-based localization shows similar or better accuracy than the baseline methods in challenging scenarios. Our results further suggest that the approach scales well and achieves even higher localization accuracy by increasing the number of learned landmarks without increasing the number of anchors.

Keywords: Localization · Mapping · Landmarks · Service Robots.

1 INTRODUCTION

Visual mapping and localization refer to creating a consistent scene representation and localizing a robot using a camera as the only exteroceptive sensor. A mobile robot's ability to localize itself in an environment is fundamental to achieve intelligent behavior. It enables a range of indoor and outdoor applications ranging from household robots (i.e., lawnmowers, vacuum cleaners) to self-driving cars.

The research in this area encompasses a broad range of methods that address this challenging task. State-of-the-art simultaneous localization and mapping (SLAM) algorithms exploit image features [22] or complete image information [4] to create sparse or semi-dense scene representation. Convolutional neural networks (CNNs) for visual localization have become an appealing alternative to the traditional methods based on hand-crafted features. In [14, 13], the authors have trained PoseNet in an end-to-end way to regress pose from single images. In contrast, there are methods [21, 5, 20] for mapping and localization that are inspired by neurobiological systems. Earlier learning approaches [5] reproduce the firing characteristics of Place- and Head-Direction Cells

[24, 28] using a hierarchical model. The model uses the concept of slow feature analysis (SFA) [30], and the intuition behind it is that behaviorally meaningful information changes on a slower timescale than the primary sensory input (e.g., pixel values in a video). Previous work [17] implemented SFA-based localization on a mobile robot. It achieved similar localization accuracy [19] to state-of-the-art visual SLAM methods, i.e., ORB and LSD-SLAM [22, 4] in small- to medium-scale environments and demonstrated robustness against changing conditions in outdoor scenarios [18, 9].

While the methods mentioned above are sufficient for the localization task, the recent trend has shifted to create semantic maps that will enable the robots to better interact with the world around them. One way to obtain such maps is to incorporate objects into the localization pipeline using deep-learning-based object detection algorithms. Recent work in this direction, i.e., Hybrid-SFA [11], uses a CNN to detect unique objects or regions in a scene and performs localization relative to them. The approach leads to representations similar to those of Spatial View Cells in the hippocampus [5]. The results show a significant improvement in localization accuracy, especially in a large-scale environment. However, it relies on hand-labeled training data to learn unique objects or regions in a scene, which is infeasible for many real-world applications.

This paper's main contribution is a novel approach that uses object instances with pre-trained visual detectors (e.g., MS-COCO objects [16]) as a labeling tool to learn new landmarks for localization. For the sake of simplicity, we will refer to objects or regions with pre-trained detectors as *anchors* and the derived objects or regions to be learned as *landmarks*. The idea is to place an anchor in spatial relation to a landmark and generate labeled training data relative to it. If the scene already contains suitable anchors, they can be used directly. Please note that it is possible to learn spatial representations directly w.r.t anchors. However, these anchors (i.e., pre-learned CNN-object categories) are typically dynamic objects in the scene (e.g., a bicycle, a chair). They thus cannot be used reliably for both indoor and outdoor localization in the long term. Hence, the proposed approach enables selecting long-term stable landmarks for localization with minimal supervision and a faster generation of labeled training data for learning them. Moreover, localization accuracy scales with the number of selected landmarks but without increasing the number of anchors and the amount of supervision.

After landmark learning, the system uses the views of the learned landmarks for mapping and localization phases. This paper presents localization results from simulated and real-world outdoor environments. Most of the available localization methods work online, but only a subset of these are trained in an offline learning phase. To provide fair and straightforward comparisons, we use PoseNet [13] and basic SFA-localization [9].

2 Related Work

The recent performance increase of deep-learning-based object detection algorithms [15] have led the way to incorporate object detection into the traditional SLAM pipeline for creating semantic maps. Earlier work [1] in this field has extended the structure-from-motion (SfM) pipeline for joint estimation of camera parameters, scene points, and object labels. However, its computational complexity limits the method's ability to operate in real-time. Other object-level SLAM methods [6, 27] use object detection in

a scene to solve the problem of scale uncertainty and drift of monocular SLAM. In [7], the authors used an extensive database of known objects and proposed an algorithm based on bags of binary words [8]. The combined usage of monocular SLAM and object recognition algorithms improves the map and finds its real scale. However, the main limitation of the approach is its dependence on known objects. QuadricSLAM [23] is an object-oriented SLAM that does not rely on prior object models. Rather it represents the objects as quadrics, i.e., sphere and ellipsoids. It jointly estimates a 3D quadric surface for each object and camera position using 2D object detections from images. In [2], the authors were the first to include the inertial, geometric, and semantic information into a single optimization framework. The proposed system performs continuous optimization over the poses while it discretely optimizes the semantic data association. In [12], the authors represented generic objects as landmarks by including an object detector in a monocular SLAM framework. The method exploits the CNN-based objects and plane detectors for constructing a sparse scene representation. The SLAM bundle adjustment includes semantic objects, plane structures, and their completed point clouds. In [29, 32], the authors use machine learning-based approaches to perform localization in indoor environments relative to landmarks. CubeSLAM [31] combines 2D and 3D object detection with SLAM pose estimation by generating cuboid proposals from single view detections and optimizing them with points and cameras using a multi-view bundle adjustment. In [25] authors create category-level models with CAD collections for real-time object-oriented monocular SLAM. Their rendering pipeline generates large amounts of datasets with limited hand-labeled data. The proposed system first learns 2D features from category-specific objects (i.e., chairs, doors) and then matches the features to a CAD model to estimate the semantic objects' pose. For obtaining a metrically correct robot pose, the system then combines semantic objects and the estimated robot's pose from VO into an optimizing graph framework. Most of the existing literature on object-SLAM considers indoor scenes or outdoor autonomous driving scenarios. In both cases, it is possible to directly use a pre-trained CNN to identify enough objects in a scene without training a detector on custom objects. However, the problem arises when a scene lacks pre-trained objects. In this scenario, training a detector on custom landmarks would automatically become a necessary pre-condition for most object-based localization approaches. While generating labeled training data for learning new landmarks by hand is cumbersome, the method described in section 3 only requires minimal human supervision for the learning task.

3 METHODS

This section introduces our proposed approach for learning new landmarks with minimal human intervention. It further presents Slow Feature Analysis (SFA), the core algorithm we use to extract spatial representation. Finally, it describes the procedure to perform localization using landmarks views.

3.1 Minimal Supervision for Landmark Learning

In this work, we propose to use readily detectable object categories from pre-trained CNNs as anchors to generate labeled data for learning new landmarks. Figure 1 shows

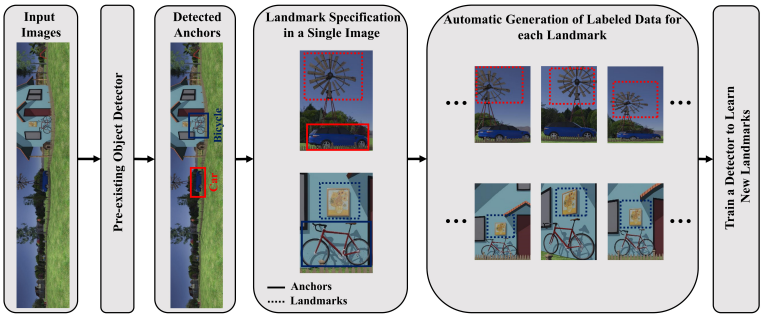


Fig. 1: **Label Generation for Learning Landmarks:** The input to the pipeline are images collected from a robot recording session. The first step applies a pre-trained visual object detector to these images to identify a pre-learned object's instances, such as a bicycle. The next step is to select new landmarks in a single image (i.e., specification of the new landmarks' spatial relationship w.r.t to the detected anchor). This selection can be made using a human input, setting a fixed offset w.r.t an anchor, or taking a random region around an anchor as a new landmark. Based on the specified relationship, the system then automatically generates labeled data from all available training images. The final step uses the generated data to train a detector for the new landmarks.

the steps of label generation and consequently using the annotated image data for landmark learning. A mobile agent explores an environment and records camera images. Here we assume that the CNN-detectable objects (e.g., a bicycle) are already present in the scene, and their location remains fixed during the recording phase. The system then runs the YOLOv3 [26] object detection algorithm on the recorded images to detect the instances of anchors. The next step is to specify the spatial relationship of new landmarks w.r.t anchors. At this step in the learning phase, minimal human supervision is necessary, i.e., the human has to specify the landmark's spatial relation. There are several possible ways to perform this step. a) A human can cooperatively indicate the location of a new landmark relative to an anchor as a 2D offset in one or a few images. This approach can generate semantic object categories (e.g., a specific tree, a fountain). b) The system autonomously analyzes the regions around the anchor and chooses a visually unique region (e.g., not a section of brick wall from a larger brick wall). c) The system takes a fixed 2D offset w.r.t to an anchor (e.g., above, bottom, besides) to learn new landmarks. We use the fixed 2D offset approach to derive a landmark relative to an anchor in this work. This offset is set only *once (i.e., minimal supervision)* in a single image for each landmark compared to manually annotating thousands of images. The system then uses the instances of detected anchors and a specified offset to automatically annotate the landmarks in the rest of the recorded images. If YOLOv3 fails to recognize the anchors in some images, we run an object tracker to obtain the bounding boxes of anchors in the missing frames. The final step uses the generated labeled data and trains a detector to recognize new landmarks. This step's output is a custom landmark detector, which we use as an independent module in the mapping and localization phases. Please note that the anchor objects in a scene are only temporarily required for landmark learning and can be removed afterwards. The basic implementation of this approach learns one landmark per detected anchor. As an extension, it is possible to scale

the system to learn multiple landmarks per anchor, which will improve both robustness against local occlusions and localization accuracy.

3.2 Slow Feature Analysis

To learn the robot's position in 2D space, we use Slow Feature Analysis (SFA) as introduced in [30]. It transforms a multidimensional time series $\mathbf{x}(t)$, in our case images along a trajectory, to slowly varying output signals. The objective is to find instantaneous scalar input-output functions $g_j(\mathbf{x})$ such that the output signals

$$s_j(t) := g_j(\mathbf{x}(t))$$

minimize

$$\Delta(s_j) := \langle \dot{s}_j^2 \rangle_t$$

with $\langle \cdot \rangle_t$ and $\dot{s}$ indicating temporal averaging and the derivative of s, respectively. The Δ-value defines the temporal variation of the output signal, and its minimization is the optimization objective. Thus small Δ-values indicate slowly varying signals over time. There are three optimization constraints: the output signals should have zero mean, unit variance, and are decorrelated. These constraints avoid the trivial constant solution and ensure that different functions g_j code for different aspects of the input.

3.3 Learning of Spatial Representation using Landmark Views

Acquiring Landmark Views: Figure 2 shows the steps to detect and extract landmark views. The input to the system are images recorded for the mapping and localization phases. The next step applies the trained detector to recognize the instances of the landmarks in images. Afterwards, we resize each landmark's bounding box to have the same size as the biggest bounding box in its category and rescale the extracted image patch to 120×120 pixels. The output of this step generates an image stream for each landmark.
Mapping Phase: We use landmark views to learn camera position regression. We choose SFA to get a compact place representation relative to each landmark and perform light-weight position regression on top. The approach employs a four-layer hierarchical SFA network and has been described recently in [11]. The network learns spatial representations relative to each landmark in an unsupervised learning process. Afterwards, we obtain metric space representation by computing a regression function from the learned spatial representations and odometry data, i.e., the robot's ground truth position (x,y). This step outputs an individual position estimator (x,y) for each landmark.
Localization Phase: The localization phase uses the learned position estimators to obtain the robot's 2D position (x,y) relative to each landmark. Afterwards, it estimates the robot's global 2D position (x,y) by combining each landmarks' position estimation using weighted averaging. We determine the weight of each landmark by taking the inverse of its localization error.

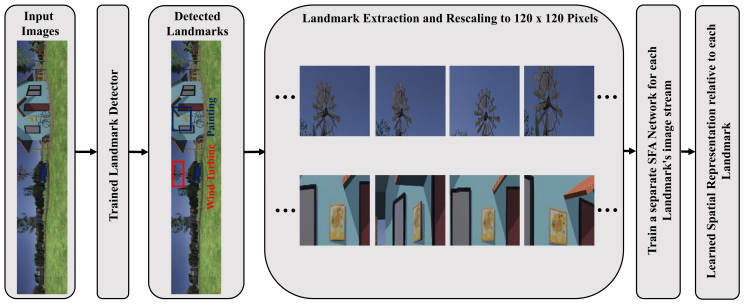


Fig. 2: **Landmark-based Learning of Spatial Representation:** We use the trained detector from the previous step for recognizing the learned landmarks in the recorded images for the mapping phase. The next step extracts the detected landmarks and rescales the image patches to 120×120 pixels. Afterwards, we train an independent SFA network to learn spatial representation relative to each landmark. The procedure is the same for the test phase (localization), where we pass each test view through its learned network for obtaining the output in the SFA space.

4 EXPERIMENTS

This section presents the localization results using simulated and real-world data. We have a two-stage system, and the first stage learns new landmarks in a scene using the proposed approach in this work. We derive one landmark per anchor by setting a fixed 2D offset. The system then uses 500 labeled images for each landmark based on the specified relationship and trains a detector to learn these landmarks. The second stage uses the learned landmarks to perform localization relative to them. This stage proves that the learned landmarks in the first stage are well suited for the localization task. To obtain baseline PoseNet [13] results, we use 25% of the data (subsampled from the training set, i.e., every 4th image) for validation and the remaining to train the network. We extract Fourier features to obtain the localization results for SFA localization on complete images, as in [9].

4.1 Simulated Experiments

We perform the experiments in a simulated garden with an area of 18×18 meters. A virtual robot randomly traverses in the environment to record images for the training set and then along a regular grid to collect the test set. We project images from the simulated omnidirectional camera to panoramic views of size 3600×600 pixels. The training and test trajectory consist of $15,000$ and $1,250$ images, respectively. Here, we have used three anchors to learn new landmarks. The anchors include a bicycle (Id_0), a car (Id_1), and an umbrella (Id_2). Table 1 shows the experimental results of localization w.r.t learned landmarks and the baseline methods. All the methods produce localization results in a similar range. However, PoseNet outperforms the SFA-based approaches in this experiment. It achieved good localization accuracy with a large amount of labeled training data, as expected in this case. However, it is infeasible to generate a massive amount of labeled training data in real-world scenarios. On the other hand, we can further improve the accuracy of landmark-based localization by incorporating more landmarks (c.f. section 4.3 on scaling experiments).

Table 1: **Localization Results on Simulated Data:** Table shows median Euclidean localization accuracy on learned landmarks views, their combination, and the baseline methods. It further reports the detection rates of the learned landmarks. The combined detection rate of 100% indicates that each test image at least contains a single landmark. PoseNet outperforms both Landmark-based and Fourier-SFA localization in this experiment.

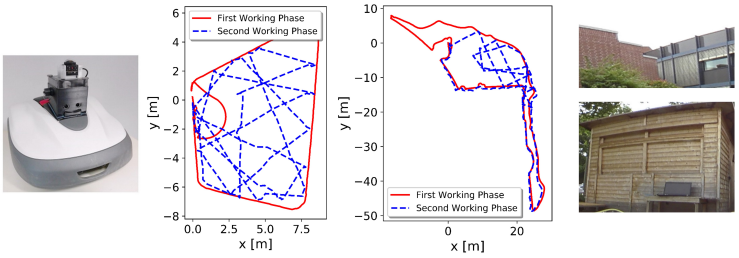
Landmark-based Localization					**Fourier-SFA**	**PoseNet**
Id_1	**Id_2**	**Id_3**	**Combined**	**%**		
0.53m [99 %]	0.37m [99 %]	0.35m [97 %]	0.33m	100	0.26m	**0.21m**

4.2 Real-World Experiments

We perform the experiments in two garden-like outdoor environments of size $88m^2$ and $494m^2$, respectively. The autonomous lawn mower robot (fig. 3a) equipped with a fish-eye lens traverses in a scene to record images of size 2880×2880 pixels. Each recording session has two operational phases. In the first phase, the robot traverses the border of an area by using the standard wire guidance technology, while in the second phase, it moves freely within the area defined by the border wire. Fig. 3b and 3c show the robot's trajectories in one of the recording sessions from each garden, respectively. During a recording, the robot stores images and the associated odometry information. For the first working phase, we estimate the robot's ground truth metric position (x,y) using a method described by Einecke et al. [3]. The authors used wheel odometry and additional weighted loop closure to get high-quality localization. However, the technique only estimates the metric shape of the boundary. For the second working phase, we estimate the ground truth data (x,y) using commercial photogrammetry software, i.e., Metashape[3]. The obtained ground truth position (x,y) estimates are used to evaluate the metric performance of the localization methods. Instead of using MS-COCO objects [16], we reuse pre-trained manually labeled region detectors (fig. 3d) from [11] as anchors to learn new landmarks for localization. We show the experimental results using ten recordings collected from both gardens under varying lighting, weather conditions, and dynamic obstacles.

Temporal Generalization This experiment aims at testing the re-localization ability of the methods in changing conditions over time. Here, we have chosen that the robot traverses on a similar path (border run) and collected three recordings from each garden. We use one dataset to learn the spatial representations and the other two sets to test the localization accuracy. The datasets differ w.r.t dynamic scene variations and changes in lighting conditions. The number of training set images for the small garden is 1138, while the two test sets consist of 1091 and 1109 images. Similarly, the big garden datasets consist of 4336 training images, while the test sets have 4032 and 4050 images. After training PoseNet, its localization accuracy on the validation data from the small and big garden is 0.07m and 0.41m, respectively. Table 2 reports median localization accuracy of landmark-based localization and the baseline methods. The results obtained with individual landmarks enable coarse localization in an environment.

[3] https://www.agisoft.com/



(a) (b) (c) (d)

Fig. 3: **Experimental Setup:** (a) An autonomous lawn mower robot with a fisheye camera used for the experiments. (b) and (c) show the robot's traversed trajectories in one of the recording sessions from both gardens, respectively. The solid line (red) shows the robot's traversal in the first working phase (border run), while the dashed line (blue) shows its traversal in the second phase (infield run). (d) We used previously trained custom objects (e.g., hut) or regions (e.g., building corner) as anchors to learn new relative landmarks for the real-world experiments.

Nevertheless, their combination achieves similar or better localization accuracy than the baseline methods. This effect, however, is more pronounced for the big garden. Fourier-SFA does well on the datasets from the small garden. However, it does not scale to the large environment with the configurations used in [9]. PoseNet produces good localization results when the environmental condition between the training and the test sets is almost identical (e.g., the first dataset from the small garden). However, it degrades otherwise (e.g., the last dataset from the big garden).

Table 2: **Real-World Experiments for Temporal Generalization:** Median Euclidean errors of individual landmarks, their combination, and the baseline methods for generalization over time. The drop in detection rates of the landmarks present in the big garden is due to their visibility only in specific parts of the scene. Please note we only use test images for the baseline methods, where at least a single landmark view was available in the corresponding image. Landmarks enable coarse localization in both environments while their combination performs similar or better than the baseline methods, especially in the big garden.

Garden	**Test Set**	**Landmark-based Localization**						**Fourier-SFA**	**PoseNet**
		Id_1	**Id_2**	**Id_3**	**Id_4**	**Combined**	**%**		
Small	1	0.26m [99 %]	0.31m [99 %]	0.60m [99 %]	-	0.20m	100	0.19m	**0.18m**
	2	0.73m [99 %]	0.93m [97 %]	1.33m [99 %]	-	**0.75m**	100	1.01m	0.83m
Big	1	1.46m [13 %]	3.54m [25 %]	1.74m [32 %]	2.44m [16 %]	**2.22m**	78	7.50m	2.99m
	2	1.83m [13 %]	3.80m [28 %]	2.06m [32 %]	2.97m [17 %]	**2.57m**	80	8.22m	6.57m

Spatial Generalization This experiment aims at testing the re-localization ability of the methods when the train and test set contain sufficiently different robot trajectories. As described earlier, each robot recording session has two operational phases. Hence, we use the images from the first phase (border run) to learn the spatial representations and the second phase (infield run) to test the localization method. We collected

two recordings from both gardens for the experiments. The small garden training sets consist of 1141 and 1131 images, while the test images from infield positions are 158 and 269, respectively. Similarly, the big garden training sets have 4336 and 4011 images, while 234 and 180 test set images. The localization accuracy of PoseNet on the validation set for both sets from the small garden is 0.06m, while the corresponding accuracy on the sets of the big garden is 0.41m and 0.30m, respectively. Table 3 reports median localization accuracy of each method. The individual landmarks again show coarse localization accuracy. However, it is slightly worse for landmarks of the big garden. Extreme perspective changes between the train and test images mainly influence this performance drop. Moreover, there is a noticeable change in the lighting conditions between the first and second robot recording phases. Despite that, their combination achieves similar or better accuracy than localization using the baseline methods. There are several ways to improve the current results of landmark-based localization. Firstly, the incorporation of more landmarks leads to a higher localization accuracy (c.f. section 4.3). Secondly, it is possible to filter out the landmarks with the worst performance as a post-processing step and only localize relative to those with mean accuracy better than a specified threshold. Thirdly, the addition of sparse images from infield run during learning can further improve the localization accuracy. Please note that, here, we only need landmark views from the infield that are relatively easy to obtain with a single pass through the pre-trained detector. It contrasts to PoseNet, which requires a computationally expensive structure-from-motion (SfM) step to generate robot poses as labeled data for learning.

Table 3: **Real-World Experiments for Spatial Generalization:** Table reports localization accuracy when the train and test sets consist of images from different robot trajectories. Similar to temporal generalization experiments, the combination of landmarks achieves similar or better accuracy than the baseline methods.

Garden	**Test Set**	**Landmark-based Localization**						**Fourier-SFA**	**PoseNet**
		Id_1	**Id_2**	**Id_3**	**Id_4**	**Combined**	**%**		
Small	1	1.41m [96 %]	0.95m [99 %]	1.01m [96 %]	-	0.74m	100	**0.66m**	0.85m
	2	0.82m [95 %]	1.05m [97 %]	1.35m [95 %]	-	0.84m	100	**0.65m**	0.82m
Big	1	4.37m [12 %]	3.95m [57 %]	1.50m [27 %]	5.51m [44 %]	**3.21m**	95	7.10m	5.31m
	2	5.88m [17 %]	4.64m [61 %]	1.82m [31 %]	6.12m [28 %]	3.96m	96	7.60m	**3.49m**

4.3 Scaling Experiments

This experiment aims to analyze the effect of increasing the number of landmarks on localization using simulated and real-world data from the small garden. We use ten different landmarks from each environment by randomly selecting them around a single anchor. The first step learns an independent position estimator for each landmark. The second step processes landmark images from the test set using the estimators and predict the robot's 2D position (x, y). Afterwards, we calculate the test set's median localization error by systematically increasing the number of landmarks. Fig. 4 shows

the results of 50 random permutations of the ten landmarks for both simulated and real-world data. Both plots initially show improved localization accuracy by increasing the number to three, and adding more landmarks continues to improve the overall localization accuracy until it almost saturates as expected. From an application perspective, a robot could increase the number of landmarks to achieve a certain accuracy level at runtime, depending on the area where high accuracy is required. As we train an independent SFA network for each landmark, the processing time will linearly increase (i.e., O(n)) with the number of landmarks. However, SFA-based mapping and localization are drastically faster than one of the fastest state-of-the-art visual localization methods [10].

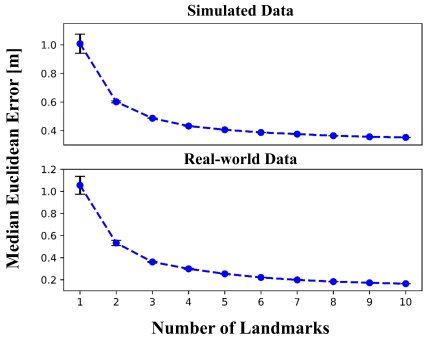


Fig. 4: **Effect of Increasing Landmarks on Localization:** Simulated data (Top). Real-world data (Bottom). The plot shows the median and variance localization accuracy for 50 random permutations of the ten landmarks. The usage of more landmarks for localization significantly improves the accuracy initially, and eventually, it saturates for a higher number of landmarks.

5 Conclusion

This work proposed an approach to speed up the label generation process for learning new long-term visual landmarks for localization. The method uses instances of readily available CNN objects as anchors to generate labeled data for the unseen imagery based on minimal human supervision. We used a fixed 2D offset to derive new landmarks relative to anchors. When anchor and landmark are within the same plane, perspective changes during recording result in labeling the identical scene part as a landmark in our training paradigm with a fixed 2D offset between anchor and landmark. In the most extreme case, if an anchor is placed such that the robot can go around it, a simple 2D approach may fail and does not capture a semantically meaningful region but a subset of the scene's viewing space. Geometry-based localization methods may fail in such a case. However, unintuitively, these views are classified very well as a landmark with a CNN, and we see no reduction in localization accuracy with pose regression learning (as shown here with SFA). After landmark learning, we used the learned landmarks and performed localization relative to them. The landmark-based localization performed better

than the baseline methods, especially in a challenging large-scale outdoor environment. The accuracy can be significantly improved by integrating more landmarks and obtaining a global position estimation relative to them. From an application perspective, our system is suitable for service robots (e.g., lawnmowers and vacuum cleaners), employing a pre-trained visual detector to learn new landmarks in a scene. Thus, the approach enables reliable localization in the long-term even if the anchor objects are no longer present in the scene.

References

1. Bao, S.Y., Bagra, M., Chao, Y., Savarese, S.: Semantic structure from motion with points, regions, and objects. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2703–2710 (2012)
2. Bowman, S.L., Atanasov, N., Daniilidis, K., Pappas, G.J.: Probabilistic data association for semantic slam. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1722–1729 (2017)
3. Einecke, N., Muro, K., Deigmöller, J., Franzius, M.: Working area mapping with an autonomous lawn mower. In: Conference on Field and Service Robotics. Springer (September 2017)
4. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-Scale Direct Monocular SLAM. In: Computer Vision - ECCV 2014, Proceedings, Part II. pp. 834–849 (2014)
5. Franzius, M., Sprekeler, H., Wiskott, L.: Slowness and Sparseness Lead to Place, Head-Direction, and Spatial-View Cells. PLoS Computational Biology **3**(8), 1–18 (2007)
6. Frost, D., Prisacariu, V., Murray, D.: Recovering stable scale in monocular slam using object-supplemented bundle adjustment. IEEE Transactions on Robotics **34**(3), 736–747 (2018)
7. Gálvez-López, D., Salas, M., Tardós, J.D., Montiel, J.: Real-time monocular object slam. Robot. Auton. Syst. **75**(PB), 435–449 (Jan 2016). https://doi.org/10.1016/j.robot.2015.08.009
8. Galvez-López, D., Tardos, J.D.: Bags of binary words for fast place recognition in image sequences. Trans. Rob. **28**(5), 1188–1197 (Oct 2012). https://doi.org/10.1109/TRO.2012.2197158
9. Haris, M., Franzius, M., Bauer-Wersing, U.: Robust outdoor self-localization in changing environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019). IEEE (2019)
10. Haris, M., Franzius, M., Bauer-Wersing, U.: Unsupervised fast visual localization and mapping with slow features. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 519–523 (2021). https://doi.org/10.1109/ICIP42928.2021.9506656
11. Haris, M., Franzius, M., Bauer-Wersing, U., Karanam, S.K.K.: Visual localization and mapping with hybrid sfa. In: Kober, J., Ramos, F., Tomlin, C. (eds.) Proceedings of the 2020 Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 155, pp. 1211–1220. PMLR (16–18 Nov 2021), https://proceedings.mlr.press/v155/haris21a.html
12. Hosseinzadeh, M., Li, K., Latif, Y., Reid, I.: Real-time monocular object-model aware sparse slam. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 7123–7129 (2019)
13. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. CoRR (2017)
14. Kendall, A., Grimes, M., Cipolla, R.: Convolutional networks for real-time 6-dof camera relocalization. CoRR (2015)

15. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems **25** (01 2012). https://doi.org/10.1145/3065386
16. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing (2014)
17. Metka, B., Franzius, M., Bauer-Wersing, U.: Outdoor self-localization of a mobile robot using slow feature analysis. In: Neural Information Processing (2013)
18. Metka, B., Franzius, M., Bauer-Wersing, U.: Improving robustness of slow feature analysis based localization using loop closure events. In: Artificial Neural Networks and Machine Learning – ICANN 2016. pp. 489–496. Springer International Publishing (2016)
19. Metka, B., Franzius, M., Bauer-Wersing, U.: Bio-inspired visual self-localization in real world scenarios using slow feature analysis. PLOS ONE **13**(9), 1–18 (09 2018). https://doi.org/10.1371/journal.pone.0203994
20. Milford, M.J., Wyeth, G.F.: Mapping a suburb with a single camera using a biologically inspired slam system. IEEE Transactions on Robotics **24**(5), 1038–1053 (Oct 2008). https://doi.org/10.1109/TRO.2008.2004520
21. Milford, M.J., Wyeth, G.F., Prasser, D.: Ratslam: a hippocampal model for simultaneous localization and mapping. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004. vol. 1, pp. 403–408 Vol.1 (April 2004). https://doi.org/10.1109/ROBOT.2004.1307183
22. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics **31**(5), 1147–1163 (oct 2015). https://doi.org/10.1109/TRO.2015.2463671
23. Nicholson, L., Milford, M., Sünderhauf, N.: Quadricslam: Constrained dual quadrics from object detections as landmarks in semantic SLAM. CoRR **abs/1804.04011** (2018)
24. O'Keefe, J., Dostrovsky, J.: The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. Brain Research **34**(1), 171 – 175 (1971). https://doi.org/https://doi.org/10.1016/0006-8993(71)90358-1
25. Parkhiya, P., Khawad, R., Murthy, J.K., Bhowmick, B., Krishna, K.M.: Constructing category-specific models for monocular object-slam. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 4517–4524 (2018)
26. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. CoRR **abs/1804.02767** (2018)
27. Sucar, E., Hayet, J.: Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 5152–5158 (2018)
28. Taube, J., Muller, R., Ranck, Jr, J.: Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. The Journal of neuroscience **10**, 420–35 (03 1990). https://doi.org/10.1523/JNEUROSCI.10-02-00420.1990
29. Thrun, S.: Bayesian landmark learning for mobile robot localization. Mach. Learn. **33**(1), 41–76 (oct 1998). https://doi.org/10.1023/A:1007554531242, https://doi.org/10.1023/A:1007554531242
30. Wiskott, L., Sejnowski, T.: Slow Feature Analysis: Unsupervised Learning of Invariances. Neural Computation **14**(4), 715–770 (2002)
31. Yang, S., Scherer, S.: Cubeslam: Monocular 3-d object slam. IEEE Transactions on Robotics **PP**, 1–14 (05 2019). https://doi.org/10.1109/TRO.2019.2909168
32. Zhao, Z., Carrera, J., Niklaus, J., Braun, T.: Machine learning-based real-time indoor landmark localization. In: Chowdhury, K.R., Di Felice, M., Matta, I., Sheng, B. (eds.) Wired/Wireless Internet Communications. pp. 95–106. Springer International Publishing, Cham (2018)