Visual Localization and Mapping with Hybrid SFA

Muhammad Haris
Frankfurt University of Applied Sciences
muhammad.haris@fb2.fra-uas.de

Mathias Franzius
Honda Research Institute Europe GmbH
Mathias.Franzius@honda-ri.de

Kaushik Sai Krishna Karanam
Frankfurt University of Applied Sciences
kaushik@stud.fra-uas.de

Ute Bauer-Wersing
Frankfurt University of Applied Sciences
ubauer@fb2.fra-uas.de

Abstract: Visual localization is a crucial requirement in mobile robotics, field and service robotics, and self-driving cars. Recently, unsupervised learning with Slow Feature Analysis (SFA) has shown to produce spatial representations that enable localization from holistic images. The approach is faster and much less complex than state-of-the-art monocular visual SLAM methods while achieving similar localization performance in small-scale environments. However, the holistic approach's performance drops significantly for highly complex, large-scale environments due to scene variations occurring during a training phase. Instead of using holistic images, an alternative is to perform localization relative to unique regions present in a scene. Therefore, in this paper, we add a new component to the SFA localization pipeline that leverages state-of-the-art CNN to identify unique image regions. Hence we propose a hybrid approach that first learns such regions with a pre-trained CNN and then uses SFA for unsupervised pose estimation relative to each region. We present the experimental results from an autonomous robot in two different outdoor environments of varying complexity and size. The experiments show the proposed hybrid approach outperforms holistic SFA w.r.t localization accuracy in both environments, but benefits are more pronounced in the large-scale environment.

Keywords: Mapping, Localization, Outdoors, Omnidirectional vision, Service robot

1 Introduction

Mapping an unknown environment and localizing within it is a vital necessity of a mobile robot. It enables a broad range of robotic applications like navigation, surveillance, and various household tasks. Localization and mapping with a monocular camera is a common choice over the other sensor modalities due to its availability, size, and cost. It is a well-researched area; hence, a variety of methods address this challenging task, ranging from feature-based methods [1, 2] to direct image-based methods [3].

The breakthrough in the performance of Convolutional Neural Networks (CNNs) for object detection [4] has allowed researchers to incorporate them into the traditional SLAM pipeline, which has led to the creation of semantically meaningful maps. One of the earliest approaches [5] in object-level SLAM has extended the structure-from-motion (SfM) pipeline to jointly estimate camera parameters, scene points, and object labels. Despite the improved results in object detection and robustness, the slower run time makes the method infeasible for real-time operation. SLAM++ [6] detects a set of known object instances and maps them with an object pose graph. Other object-level SLAM approaches [7, 8] model objects as spheres to overcome the scale drift problem of monocular SLAM. Dorian Galvez et al. [9] considered an algorithm based on bags of binary words [10] that leverages a massive database of objects. To improve the map and find the real scale, the monocular SLAM algorithm and the object recognition algorithm together exploit not only the object rigidity constraints but also the earlier observations acquired by SLAM that serves as cues for the location of

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.

the objects in the present image. This approach leads to faster and more detections, thus it provides more geometrical constraints to SLAM. However, the method does not show its ability to work with dynamic objects. QuadricSLAM [11] is an online object-oriented SLAM system that does not use prior models and represents objects as quadrics, i.e., sphere and ellipsoids, thus approximating an object's position, orientation, and shape. Bowman et al. [12] use semantic objects e.g., chairs and doors, in a semantic SLAM approach. The proposed system continuously optimizes the pose while it discretely optimizes the semantic data association. Therefore, it divides the metric semantic SLAM into two sub-problems. The system couples the inertial, geometric, and semantic information into a single optimization framework. Fusion++ [13] is another object-level SLAM system that focuses on indoor scene understanding with an RGB-D camera. The system produces semantically labeled Truncated Signed Distance Function (TSDF) reconstructions of the objects with the Mask-RCNN object detector. Afterward, the system adds the TSDF object instances to the map for tracking, graph optimization, and relocalization. Hosseinzadeh et al. [14] include an object detector in a monocular SLAM framework to represent generic objects as landmarks. Although camera localization is quite precise with sparse point-based SLAM, it lacks semantic information. CNN-based object detectors perform well in providing essential information about the objects from single images. Therefore, this method uses CNN based object and plane detectors to construct sparse semantic map representation for localization. Semantic objects and plane structures, along with their completed point clouds, are included in the SLAM bundle adjustment. However, the method requires a post-processing step to recover the scale of the inserted objects, which is both expensive and error-prone. CubeSLAM [15] 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 multi-view bundle adjustment. Parkhiya et al. [16] construct category-level models with CAD collections for real-time object-oriented monocular SLAM. The authors develop a rendering pipeline that helps in generating large amounts of datasets with limited hand-labeled data. The proposed system first learns 2D features from category-specific objects, such as chairs and doors with a deep network. The next step then matches the learned features to a 3D CAD model for pose estimation of the semantic object. The final step adds these semantic objects and the estimated robot's pose from VO to an optimizing graph framework for obtaining a metrically correct robot pose.

In contrast to other methods, the bio-inspired model for SFA-based localization [17, 18] learns spatial representations of an environment in an unsupervised way. The model uses the concept of slow features analysis (SFA) [19], and the intuition behind it is that behaviorally meaningful information changes on a slower timescale compared to the primary sensory input (e.g., pixel values in a video). The usage of holistic images and agent movement statistics in these models led to spatial representations similar to grid cells and place cells, whereas our approach here leads to representations similar to those of spatial view cells [20]. Previous work [17] implements SFA-based localization on a real robot using holistic views in an outdoor environment. It establishes a system that learns instantaneous representations of the robot's position in an unsupervised learning process. It achieves similar or equal performance compared to state-of-the-art visual SLAM methods, i.e., ORB and LSD-SLAM [2, 3] in a small-scale environment for different test scenarios. In this work, we follow the object-oriented SLAM paradigm and include a CNN-based object detector to the SFA localization pipeline. Most of the object-oriented SLAM methods so far have been demonstrated on the data collected from indoors or car cameras. In both scenarios, the pre-trained networks like YOLOv3 [21] provide a rich set of object identifications without retraining. However, we perform the experiments in garden-like environments that do not typically contain the pre-trained object categories. To the best of our knowledge, only in [22], the authors perform robot experiments in a garden environment. However, they use ArUco markers for localization, which may not be feasible for many real-world applications as it requires scene modification. Hence, we here retrained YOLOv3 [21] to identify pre-selected unique regions that are present in an unstructured outdoor environment. The pre-selected regions serve as landmarks but are not necessarily objects. A garden often contains many non-unique objects. Here, we choose unique scene parts composed of multiple partial objects. Most regions with a minimum visual variability (e.g., not a white wall) and stability (e.g., not a patch of sky) can serve this purpose. To summarize, our approach is quite generic, as retraining the deep neural network for custom objects is optional. In a scenario where a service robot has a built-in object detection module, and the environment contains pre-trained objects, the proposed localization approach can enable fast mapping and localization with a straight-forward model.

Section 2 provides the details of hybrid SFA mapping and localization pipeline. For the sake of completeness, it also presents SFA and its mathematical definition. Section 3 describes the experimental

setup, procedure, and the results for localization in a real-world setting from two different outdoor environments. Section 4 concludes the work and outlines the follow-up work.

2 Hybrid SFA

Previous work [17] demonstrates the successful implementation of SFA-based localization on an outdoor robot with holistic views. In this work, instead of using holistic views for localization, the aim is to use small image parts, i.e., unique regions for this task. Therefore, in hybrid SFA, we employ a CNN-based selection step for supervised learning of such regions and separately process them to extract spatial representations with SFA. Finally, we combine the individual position (x, y) estimates from each detected region in a post-processing step to obtain a test image's global 2D position (x, y).

CNN-based Unique Region Selection The core idea of the proposed approach is to perform localization relative to unique regions in a scene. Therefore, as a first step, we fine-tuned the state-of-the-art YOLOv3 [21] pre-trained on the ImageNet database [23] with our hand-labeled data to detect those regions (fig. 1a). Please note that the scene-specific training was necessary because the environments in which we performed the experiments do not contain the pre-trained objects; otherwise, this step may be omitted.

Slow Feature Analysis for Learning Spatial Representations Slow feature analysis (SFA) introduced in [19] 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.
For learning a spatial representation with SFA, the goal is to find functions representing the robot's position on the x- and y-axis as slowly varying features while being independent of other variables like its orientation or condition changes. SFA-based learning depends on temporal statistics of the input data; therefore, if the relevant information (here: robot position) during training changes on a slower timescale compared to other variables, it will be encoded as slowest features in the learned representations. The work in [20] provides a detailed mathematical analysis of SFA for localization based on the inputs' temporal statistics.

Mapping Pipeline Fig. 1b shows our proposed pipeline to map an environment. The input to it is an image stream collected from a robot recording. It uses the learned detector to identify unique regions from the images. After the detection phase, a subsequent post-processing step extracts each unique region from the images and rescales them to a fixed resolution of 120×120. At the end of this step, we have a separate image stream for each unique region. The next step independently processes images of those regions with a four-layer hierarchical SFA network to extract slow features. The network parameter settings are the same as already published in [17]. However, we adapted some specific settings to account for a different image resolution. The layers in the network consist of multiple SFA nodes arranged on a regular grid. Each node performs two steps, i.e., linear SFA followed by quadratic SFA. Linear SFA can find better solutions with a nonlinear expansion of its inputs, similar to nonlinearities in neural networks. However, this expansion increases feature dimensionality. Therefore, the first step performs dimensionality reduction with linear SFA for computational efficiency. Finally, the second step projects the features into the space of quadratic

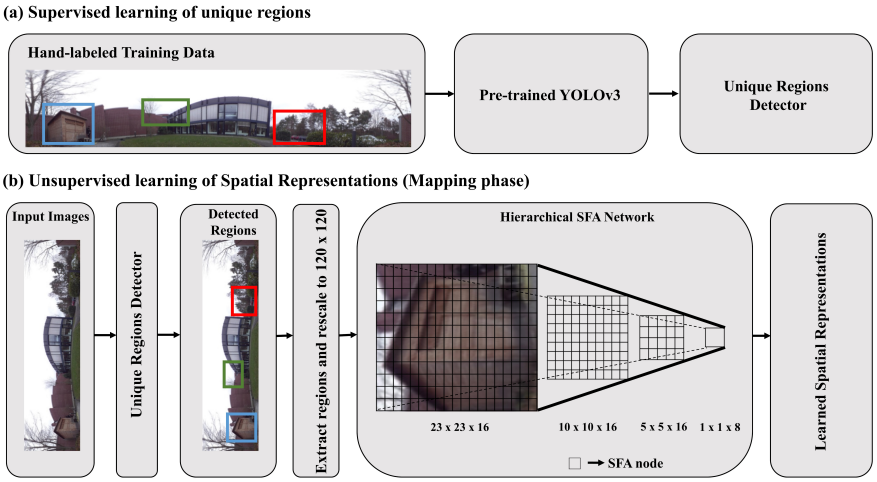


Figure 1: **Mapping pipeline:** (a) The first step uses hand-labeled image data for supervised learning of unique regions in a scene with state-of-the-art YOLOv3 [21] algorithm. We use a pre-trained network and fine-tune it with only a few hundred examples. The output of this step is a custom detector for recognizing unique regions. (b) The second step makes use of detected image regions for learning spatial representations. The procedure starts with detecting and extracting regions from the input images followed by a rescaling step, which rescales them to 120×120 resolution. The next step independently processes each region using a four-layer hierarchical SFA network resulting in a separate SFA-model for each image region. The layers in the network consist of multiple SFA nodes. Each node performs linear SFA for dimensionality reduction followed by the application of SFA on quadratically expanded inputs for slow feature extraction. The final node outputs an eight-dimensional feature vector, i.e., the slowest features $s_{1...8}$ for each image region identified at a position (x, y). The output of the pipeline is the learned spatial representation relative to each unique region.

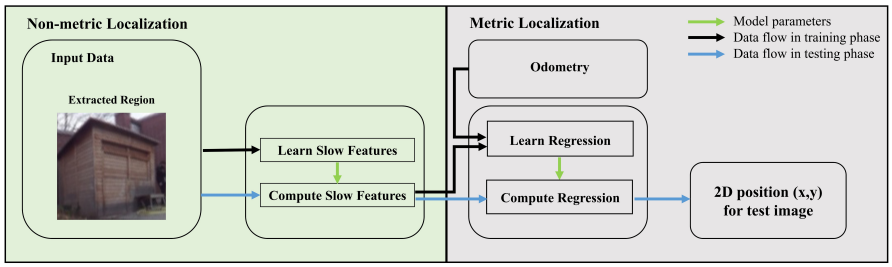
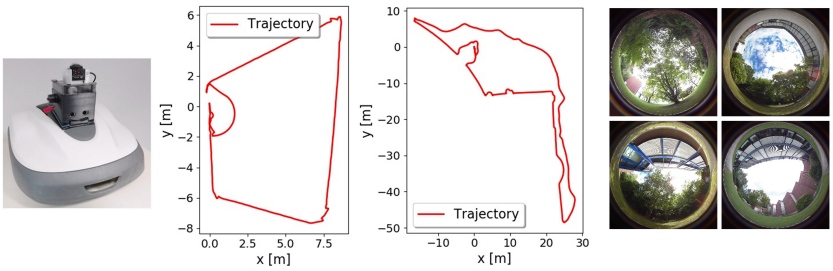


Figure 2: **Localization pipeline:** SFA-based localization is instantaneous; i.e., it only needs a single image to compute the model output. Therefore, it takes an extracted image region from a test image and uses its associated trained-SFA model to compute its representation in the SFA space. To quantify the model performance in metric space, we use odometry data and learned SFA representations to compute a metric regression function. Finally, we apply the function on the SFA output of a test image to obtain its 2D (x, y) position. Please note this mapping from SFA to metric space only serves the purpose of metric evaluation.

monomials and then applies SFA to extract slow features. The final node in the network outputs the first eight slowest features $s_{1...8}$ for each input image. Please note that, at this step, we obtain a separate SFA model trained with the views of each detected region. The output of the pipeline is the learned spatial representation relative to each unique region.

Non-metric vs. Metric Localization The left part (green) of the localization pipeline (fig. 2) shows the procedure to obtain a test image's output in SFA space, i.e., non-metric localization. The input to the pipeline is the extracted unique region from a test image. We process the region's view



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

Figure 3: **Experimental setup:** (a) A modified lawn mower robot equipped with an upright fisheye camera is used for the experiments. (b) and (c) show the traversed trajectory by the robot in the small-scale and large-scale environment, respectively. (d) shows example images from the two outdoor environments.

with its associated trained SFA model to compute its output. Please note that non-metric localization is sufficient for SFA-based navigation [24, 25]. In the fig. 2, we only show one region; the procedure is the same for all unique regions extracted from a single test image.
The right part (gray) of the localization pipeline (fig. 2) shows how to obtain a test image's output in metric coordinates (x, y). Please note that this step is required only for assessing the learned spatial representation in metric space. We project SFA space to the metric space by computing a regression function with learned representations and odometry data. Finally, we compute the 2D position (x, y) of a test image by applying the regression function. The procedure is also the same for all extracted regions from a single test image.

Global Position Estimate As mentioned earlier, we process detected regions separately and obtain individual 2D position (x, y) estimates for each test image. The combination of these estimates to get a global 2D position (x, y) depends on the application scenario (c.f. section 3, Table 1). In the small-scale environment, it is possible to identify most regions in images from most positions. Therefore, we concatenate SFA outputs of individual regions and compute a single regression function to obtain the combined position estimate (x, y) for a test image. In contrast, in the large-scale environment, a particular region is visible only in a specific part of the scene. Hence, we here combine the individual regression estimates by averaging them only for images with more than one detected region.

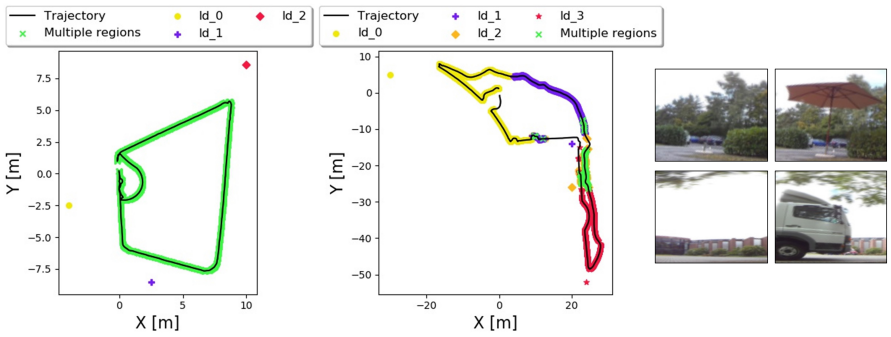
3 Experimental Results

We perform all the experiments in a small- and a large-scale garden-like environment. The small garden work area is $88m^2$, while the big garden work area is $494m^2$. We use a modified lawn mower robot equipped with an upright fisheye camera. It captures images of size 2880×2880 pixels. Figure 3 shows the robot, the trajectories traversed by the robot during recordings in both gardens, and example images from our dataset. During each recording, the robot autonomously follows the border wire buried in the ground using the standard wire guidance technology while storing omnidirectional views and the associated odometry information. Please note that it is possible to obtain highly precise position (x, y) estimates using loop closures and subsequent error correction [26] for the border run, which we use as the ground truth data for metric evaluation of the localization methods. We collected three different recordings from each garden, which differs w.r.t illumination changes like sunny vs. overcast conditions and dynamic obstacles.
The learning of spatial representations relative to unique regions in a scene requires training a detector to recognize those regions. Therefore, we train YOLOv3 [21] for this task using one of its implementations [27]. Please note that we use images from different recordings to train the detector, i.e., the training set does not include images from recordings used for the localization task. An initial preprocessing step unwraps omnidirectional images of size 2880×2880 pixels to corresponding panoramic views of size 1200×200 pixels. Afterwards, we hand-labeled three unique regions for the small garden and four unique regions for the big garden using Microsoft's visual object tagging tool [1]. The number of labeled images are 650 and 750 for the small and big garden, respectively.

[1] https://github.com/microsoft/VoTT

Table 1: This table shows the detection rate of each learned region with state-of-the-art YOLOv3 [21]. Due to the smaller area, all the learned regions are detectable in an entire image sequence from the small garden. In contrast, most of the time, only a single unique region is detectable in a specific part of the big garden. A significant drop in the region's (Id_0) detection rate for the third recording of the big garden is due to occlusions. There were no false positives detected for the small garden, while for the big garden, we filtered out the false positives based on the confidence score threshold of 0.5.

Environment	**Recording**	**Detection Rate [%]**			
		Id_0	**Id_1**	**Id_2**	**Id_3**
Small Garden	**1**	88	96	99	-
	2	87	95	99	-
	3	88	93	99	-
Big Garden	**1**	34	22	16	31
	2	32	19	15	32
	3	18	19	16	30



(a) (b) (c)

Figure 4: **Detection of unique regions:** (a)We hand-labeled three unique regions in the small garden. Due to its size, it is possible to detect each region at all positions in the trajectory. Green color shows positions with more than one detected region. (b) For the big garden, we hand-labeled four unique regions. It is apparent from the figure that only a small percentage of positions contain more than one detected region (green), while most of them contain a single detected region. (c) Some example images from the training set (left) and a test set (right) show scene variation caused by a dynamic object. For the second example, the dynamic object mostly occludes the unique region in the scene.

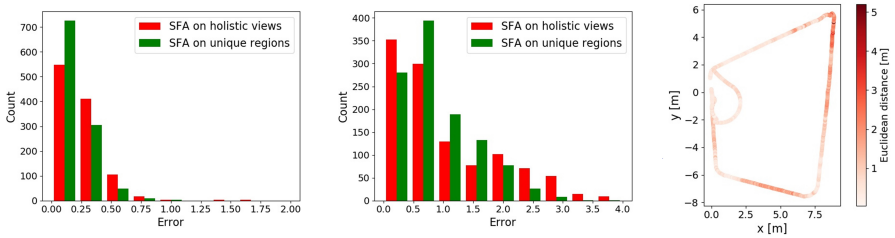
After the training phase, we used the learned detector to identify the regions in the test sets. Table 1 presents the detection rate of each learned region, and figure 4 visualizes the detections for one of the recordings in both gardens. Afterwards, we follow the same procedure as described in section 2 for training SFA network, obtaining individual 2D position (x, y) estimates, and combining them to get a global 2D position (x, y) for a test image. To obtain the results for holistic SFA, we use the architecture with the same parameter settings and procedure to obtain the 2D position (x, y) of a test image, as published in [17].

3.1 Small Garden

We use one recording for learning spatial representations and two different recordings to test the localization performance. The first test set has more similar conditions, w.r.t the training set than the second set, which has more different illumination conditions and dynamic objects. Table 2 shows the localization results based on each unique region, their combination, and holistic images. The results show that it is possible to enable localization even with a single unique region, and their combination achieves higher accuracy than localization based on holistic views. The performance degradation of certain regions, for instance, region (Id_1) for the second test set, is due to the dynamic objects that were not present in the training set (see fig. 4c).

Table 2: **Localization Results in the Small Garden:** This table presents mean localization performance with SFA on unique regions, their combination, and for holistic views. The proposed approach allows localization from a single detected region in a scene. However, combining multiple regions for this task outperforms the established holistic SFA w.r.t localization accuracy. The significant performance drop in the second test set for individual regions is due to scene variation between the training and test set images, i.e., illumination changes and dynamic objects. Please refer to fig. 5 for error distribution of both test sets for combined and holistic SFA.

Experiment	**SFA on Unique Regions [m]**				**Holistic SFA [m]**
	Id_0	**Id_1**	**Id_2**	**Combined**	
Test Set 1	0.30	0.38	0.68	**0.20**	0.25
Test Set 2	0.79	2.47	1.65	**0.90**	1.04



(a) (b) (c)

Figure 5: **Error distribution and visualization (small garden):** (a) and (b) show the error distribution of the two test sets, respectively. (a) Test Set 1: mean localization error for SFA on unique regions and holistic views is 0.2 m and 0.25 m, respectively. (b) Test Set 2: mean localization error for SFA on unique regions and holistic views is 0.9 m and 1.04 m, respectively. (c) shows the visualization of errors for the second test set estimated from SFA on unique regions.

3.2 Big Garden

Like the small garden experiments, we use one recording for learning the spatial representations and two different sets for evaluation. However, one of the main differences is that this garden is much bigger and more complex than the small garden. Moreover, the learned regions are visible only in specific parts of the garden; thus, localization w.r.t only one unique region is possible for the most part. Table 3 shows the localization results based on four different unique regions, their combination, and holistic images. The results show a drastic improvement in localization performance over holistic SFA for both unique regions and their combination. The high scene variation due to several factors like lighting conditions and dynamic objects, even within a single recording, makes it difficult to encode spatial representations with holistic images. In contrast, focusing on specific regions for learning spatial representations simplifies the learning process, thus resulting in better localization performance. However, the occlusions caused by dynamic objects reduces the localization performance of individual regions.

Table 3: **Localization Results in the Big Garden:** This table presents mean localization performance with SFA on unique regions, their combination, and for holistic views. The localization relative to unique regions significantly improves the performance over the holistic image localization with SFA. The higher scene variation affects the learned spatial representations with complete images. The proposed approach allows the mitigation of such variations by focusing on a smaller region of a scene hence achieving better localization performance. Please refer to fig. 6 for error distribution of both test sets for combined and holistic SFA.

Experiment	**SFA on Unique Regions [m]**					**Holistic SFA [m]**
	Id_0	**Id_1**	**Id_2**	**Id_3**	**Combined**	
Test Set 1	2.75	1.94	1.59	1.67	**2.05**	7.99
Test Set 2	4.95	2.22	0.70	1.92	**2.48**	8.42

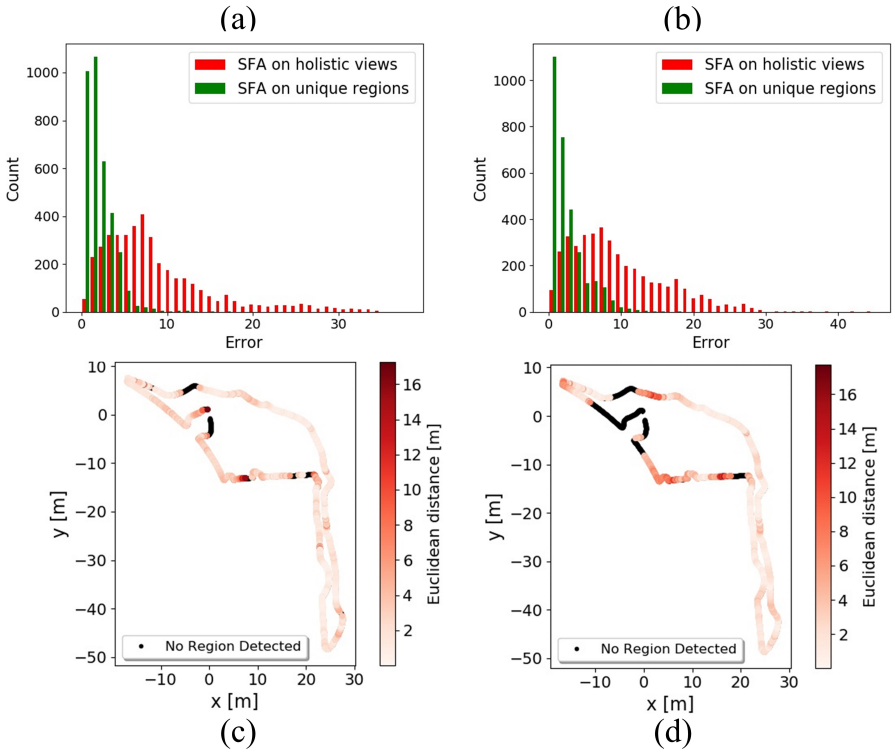


Figure 6: **Error distribution and visualization (big garden):** (a) and (b) show the error distribution of the two test sets, respectively. (a) Test Set 1: mean localization error for SFA on unique regions and holistic views is 2.05 m and 7.99 m, respectively. (b) Test Set 2: mean localization error for SFA on unique regions and holistic views is 2.48 m and 8.42 m, respectively. (c) and (d) show the visualization of errors for the two test sets estimated from SFA on unique regions.

4 Conclusion

In this work, we introduced a new approach for SFA-based localization that uses state-of-the-art CNN to learn unique regions in a scene and perform localization relative to them. We presented the experimental results from robot recordings in two different outdoor environments. The results show that the new method significantly outperforms holistic-SFA in a large-scale environment and achieves similar results in a small-scale environment. Scene variation in the training data influences the learning of spatial representations with holistic-SFA. However, focusing on smaller image parts, i.e., unique regions, helps to reduce such effects. It leads to better encoding of spatial representations hence achieves higher localization accuracy. The occlusions produced by dynamic objects affect the localization performance based on individual unique regions. However, incorporation of more landmarks for localization would help to overcome this problem. In this work, we assume unique regions are guaranteed by human preselection. Nevertheless, this step is not a prerequisite for our proposed approach. If the scene contains enough pre-trained CNN objects, this step is optional. For fully unsupervised operation, pre-trained object categories may be used alternatively (e.g., category "bicycle" from COCO [28] training set) if uniqueness within the scene is to be validated by the robot. A difference from conventional landmark-based localization approaches, for instance, triangulation, is that our system enables localization from a single image region. In contrast, triangulation requires the simultaneous detection of at least three landmarks with a known position. Moreover, end-to-end approaches for 6D pose estimation like PoseNet [29] requires expensive SfM modeling for labeled data, whereas our system operates with simple odometry. As shown for the small garden, localization accuracy can be significantly increased by training more unique region detectors and learning SFA representations. Our system can scale between low computational cost and low localization accuracy to higher accuracy at a modestly higher computational cost.

References

[1] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008. doi: 10.1177/0278364908090961.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, oct 2015. ISSN 1552-3098. doi:10.1109/TRO.2015.2463671.

[3] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision - ECCV 2014, Proceedings, Part II*, pages 834–849, 2014.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[5] S. Y. Bao, M. Bagra, Y. Chao, and S. Savarese. Semantic structure from motion with points, regions, and objects. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2703–2710, 2012.

[6] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[7] D. Frost, V. Prisacariu, and D. Murray. Recovering stable scale in monocular slam using object-supplemented bundle adjustment. *IEEE Transactions on Robotics*, 34(3):736–747, 2018.

[8] E. Sucar and J. Hayet. 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)*, pages 5152–5158, 2018.

[9] D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel. Real-time monocular object slam. *Robot. Auton. Syst.*, 75(PB):435–449, Jan. 2016. doi:10.1016/j.robot.2015.08.009.

[10] D. Galvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *Trans. Rob.*, 28(5):1188–1197, Oct. 2012. doi:10.1109/TRO.2012.2197158.

[11] L. Nicholson, M. Milford, and N. Sünderhauf. Quadricslam: Constrained dual quadrics from object detections as landmarks in semantic SLAM. *CoRR*, abs/1804.04011, 2018.

[12] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic slam. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, 2017.

[13] J. Mccormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric object-level slam. pages 32–41, 09 2018. doi:10.1109/3DV.2018.00015.

[14] M. Hosseinzadeh, K. Li, Y. Latif, and I. Reid. Real-time monocular object-model aware sparse slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7123–7129, 2019.

[15] S. Yang and S. Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, PP:1–14, 05 2019. doi:10.1109/TRO.2019.2909168.

[16] P. Parkhiya, R. Khawad, J. K. Murthy, B. Bhowmick, and K. M. Krishna. Constructing category-specific models for monocular object-slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4517–4524, 2018.

[17] B. Metka, M. Franzius, and U. Bauer-Wersing. Bio-inspired visual self-localization in real world scenarios using slow feature analysis. *PLOS ONE*, 13(9):1–18, 09 2018. doi:10.1371/journal.pone.0203994.

[18] B. Metka, M. Franzius, and U. Bauer-Wersing. Outdoor self-localization of a mobile robot using slow feature analysis. In *Neural Information Processing*, pages 249–256. Springer Berlin Heidelberg, 2013.

[19] L. Wiskott and T. Sejnowski. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Computation*, 14(4):715–770, 2002.

[20] M. Franzius, H. Sprekeler, and L. Wiskott. Slowness and Sparseness Lead to Place, Head-Direction, and Spatial-View Cells. *PLoS Computational Biology*, 3(8):1–18, 2007.

[21] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. URL http://arxiv.org/abs/1804.02767.

[22] N. Strisciuglio, M. L. Vallina, N. Petkov, and R. M. Salinas. Camera localization in outdoor garden environments using artificial landmarks. In *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pages 1–6, 2018.

[23] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[24] B. Metka, M. Franzius, and U. Bauer-Wersing. Efficient navigation using slow feature gradients. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017*, pages 1311–1316, 2017.

[25] M. Haris, M. Franzius, and U. Bauer-Wersing. Robot navigation on slow feature gradients. In *Neural Information Processing*, pages 143–154. Springer International Publishing, 2018. ISBN 978-3-030-04239-4.

[26] N. Einecke, K. Muro, J. Deigmöller, and M. Franzius. Working area mapping with an autonomous lawn mower. In *Conference on Field and Service Robotics*. Springer, September 2017.

[27] A. Muehlemann. Trainyourownyolo: Building a custom object detector from scratch, 2019. URL https://github.com/AntonMu/TrainYourOwnYOLO.

[28] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

[29] A. Kendall, M. Grimes, and R. Cipolla. Convolutional networks for real-time 6-dof camera relocalization. *CoRR*, 2015. URL http://arxiv.org/abs/1505.07427.