Spatial Perception by Object-Aware Visual Scene Representation

Chung-Yeon Lee[1,2]
cylee@bi.snu.ac.kr

Hyundo Lee[1]
hdlee@bi.snu.ac.kr

Injune Hwang[1]
ijhwang@bi.snu.ac.kr

Byoung-Tak Zhang[1,2]
btzhang@bi.snu.ac.kr

[1]Seoul National University, Seoul, South Korea
[2]Surromind Robotics, Seoul, South Korea

Abstract

Spatial perception is a fundamental ability necessary for autonomous mobile robots to move robustly and safely in the real-world. Recent advances in SLAM enabled a single camera-based system to concurrently build 3D maps of the world while tracking its location and orientation. However, such systems often fail to track themselves within the map and cannot recognize previously visited places due to the lack of reliable descriptions of the observed scenes. We present a spatial perception framework that uses an object-aware visual scene representation to enhance the spatial abilities. The proposed representation compensates for aberrations of conventional geometric scene representations by fusing those representations with semantic features extracted from perceived objects. We implemented this framework on a mobile robot platform to validate its performance in home situations. Further evaluations were conducted with the ScanNet dataset which provides large-scale 3D photo-realistic indoor scenes. Extensive tests show that our framework can reliably generate maps by reducing tracking-failure, and better recognize overlap in the map.

1. Introduction

Spatial perception is the ability to be aware of the spatial relationships with respect to the position of one's body despite distracting information [10]. Reliable navigation, object manipulation, autonomous surveillance, and many other tasks of spatial AI systems from mobile robots to self-driving vehicles require accurate, robust, and fast spatial abilities [6]. Hence, the research areas of spatial perception such as structure from motion (SfM), visual odometry, and simultaneous localization and mapping (SLAM) have drawn considerable attention from robotics, computer vision, and AI communities [22, 2, 21, 8].

The underlying representation of scene geometry is a crucial element of such spatial perception frameworks since they usually map a cloud of points using geometric primitives including points, lines, patches, and non-parametric surface representations. In particular, visual SLAM was traditionally introduced as a method for tracking geometric keypoint descriptors along successive frames and then minimizing an objective function based on reprojection errors to estimate the mobile agent's poses [5, 7].

Since geometric representations have advantages of being directly observed by visual sensors and measured based on continuous quantities, they have been used to generate a map of the real-world. However, geometric representations are limited when it comes to bad situations having a lack of reliable descriptions of the observed scenes. This limitation makes a spatial perception system tend to fail to localize itself on the map, hence becoming unable to correctly recognize previously visited places [26].

To perceive spatial layouts based on extremely partial and ambiguous cues, human spatial abilities have evolved to give attention to semantic contents as representations of geometrical uncertainties [17]. For example, our brain continuously assigns values to distinguishable parts such as objects and human faces, forming a priority or a saliency map of the visible space by saccades [11, 9]. Thus, although our eyes receive incomplete and uncertain information, the spatio-visual perception system in our brain is usually able to construct a stable representation of the world successfully [23].

Recently, some semantic mapping approaches also have focused on using semantic representations. Salas-Moreno et al. proposed the SLAM++ system which trains domain-specific object detectors corresponding to repeated objects like tables and chairs. The trained detectors are then integrated into the SLAM framework to recognize and track those objects resulting in the semantic map [20]. Choudhary et al. also worked on object discovery in the maps for the purpose of closing loops [3]. Sünderhauf et al. developed a method for object-oriented semantic mapping, where individual object instances are critical entities in the maps [24]. Tateno et al. proposed a novel framework that integrates depth prediction based on the convolutional neural networks (CNN) into the SLAM system. They fused semantic segmentation labels with the global 3D model to obtain dense depth maps along texture-less surfaces [25]. However, the majority of the semantic mapping approaches are still limited to mapping objects that are present in a predefined database and attempt to retrieve a dense description of the environment at the overly cost.

To cope with such limitations and enhance the spatial abilities of the autonomous driving system, we present an object-aware visual feature augmentation framework which utilizes semantic features to augment sparse geometric visual features instead of the previous approaches that completely describe the environment. In our framework, the semantic features extracted from objects observed in the environment compensate for aberrations derived from geometric features. Therefore, they perform a crucial role in core processes such as feature matching and relocalization, which are widely implemented in the spatial perception systems.

In the rest of this paper, we first explain our proposed framework, and then quantitative and qualitative evaluations proves that the framework is beneficial for reliable mapping and better recognition of the overlap in the map. Finally, we discuss the experimental results and insights obtained.

2. Methods

The underlying spatial perception system we use is based on the visual SLAM which is widely used in autonomous robots and virtual reality applications. Conventional visual SLAM systems rely on the extraction of geometric local features from image frames to carry out a sparse reconstruction of the observed scene and to estimate the camera pose.

In this work, we implement our framework into the visual SLAM system by adding a feature augmentation module that fuses the geometric feature with the corresponding semantic feature which is a distribution of per-pixel object-class probabilities estimated based on the CNN-based object detection algorithm.

The implemented module can be applied directly to visual SLAM processes including feature matching and relocalization, as depicted in Figure 1. Specifically, the main contributions of this framework lie in the semantic-augmented visual vocabulary which is an improved scene representation based on similarity measure of descriptors and an efficient selection algorithm for keyframe candidates.

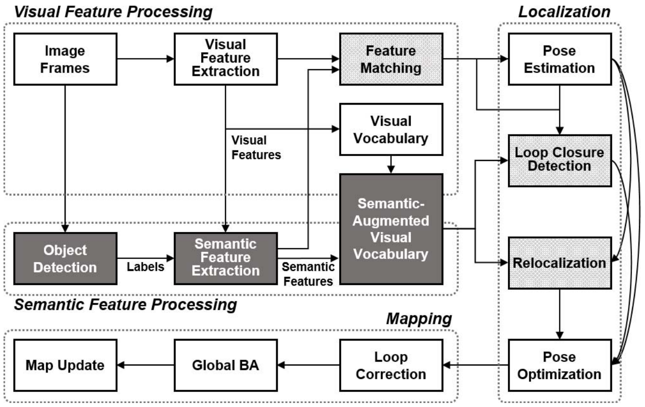


Figure 1: System overview

2.1. Semantically Augmented Scene Representation

The spatial features of the traditional visual SLAM consist of a keypoint and a descriptor. The keypoint is a distinguishable point usually detected on edges or corners, and the descriptor is highly distinctive and partially invariant value which represents the keypoint. To classify informative frames, the visual SLAM system also uses a bag-of-words model-based visual vocabulary generated by a distribution of frequently observed descriptors. Our proposed module augments such descriptors and visual vocabulary vectors by adding semantic features.

For each image frame f, let L_f denote a set of keypoint labels, each of which has the highest probability value among the semantic features for the corresponding keypoint, and P_f^l be a set of keypoints with label l. Similarly, let B_f and P_f^v denote a set of visual vocabulary indices and a set of keypoints with the index v of the visual vocabulary, respectively.

As in typical bag-of-words models, we define a weight ω as a number of occurrences of either the semantic label or the visual vocabulary index.

The weight for keypoint label l is proportional to the number of keypoints in frame f with label l. Similarly, for each visual vocabulary index v, the weight is proportional to the number of keypoints with visual vocabulary index v in frame f. Consequently, the augmented scene representation at frame f can be defined as follows:

$$\begin{aligned} \mathbb{L}_f := & \left\{ \langle l, \omega \rangle \mid l \in L_f,\ \omega = \gamma |P_f^l| \right\} \\ & \cup \left\{ \langle v, \omega \rangle \mid v \in B_f,\ \omega = \eta_v |P_f^v| \right\}, \end{aligned} \quad (1)$$

where γ is a constant factor for labels that balances the influence of semantic features at a similar level to visual features, and η_v is a factor for visual vocabulary depending on the frequency of visual vocabulary determined in advance.

2.2. Improved Similarity Measure of Descriptors

Descriptor distance, which measures the similarity between descriptors, plays an essential role in visual SLAM since the system is basically a feature matching system. When comparing two keypoints in different frames, without further information, the fact that their labels are the same does not guarantee that they are identical since they might just be the two different points on the same object. Discrepancy in labels, in contrast, can be seen as a clear evidence that the keypoints do not match. In this sense, a distance factor δ between descriptors p and q, with a penalty factor ξ greater than 1, can be defined as follows:

$$\begin{aligned} dist(p,q) :=& \delta_{pq} \cdot \|p,q\|_n \, , \\ \delta_{pq} :=& \begin{cases} 1 & \text{if } l_p = l_q \\ \xi & o.w. \end{cases} \end{aligned} \tag{2}$$

where l_p and l_q are labels of p and q, respectively, and $\|p,q\|_n$ is an arbitrary distance between p and q.

The distance factor δ_{pq} is then multiplied to $\|p,q\|_n$ in order to update the distance in a way that considers perceived objects. This update method is not restricted to certain types of distance, and thus applicable to arbitrary visual perception systems based on any distance metrics.

2.3. Efficient Selection Algorithm for Keyframe Candidates

When tracking fails due to rapid camera motion, occlusion or motion blur, it is necessary to compute the camera pose with respect to the map again [27]. This process is called a relocalization, and it is crucial for any spatial perception system because tracking-failure causes a chaotic map and several risks associated with the map.

In this process, selecting an appropriate number of keyframes to be matched is important because keypoint matching for all frames is highly inefficient. Here, we can use the semantic-augmented visual vocabulary described in Section 2.1 for selecting the candidate keyframes by using Algorithm 1. In the proposed algorithm, we represent 3 steps as Equation 3 to 5 while selecting the final candidates.

C_f^1 is a set of pairs of a keyframe k containing the same labels as f, and n, the number of such labels. C_f^2 is a set of keyframes such that for the pair $\langle k, m\rangle \in C_f^1$ containing each keyframe k, m is greater than the maximum n of C_f^1 multiplied by constant $\alpha < 1$. C_f^3 is a set of pairs consisting of a keyframe k that has the maximum L1-norm of f, and a, the summation of the L1-norm.

Algorithm 1 Candidate keyframe selection

```
Input : current frame $f$, map $M$
Output : candidate keyframe list
initialize $C1$, $C2$, $C3$, $Candidates$
for each $k \in$ all keyframes in M do
    $n \leftarrow$ number of labels in both $f$ and $k$
    if $n \neq 0$ then
        add $\langle k, n\rangle$ to $C1$
    end if
end for
for $\langle k, m\rangle \in C1$ do
    if $m > \alpha\cdot$(max $n$ in $C1$) then
        add $k$ to $C2$
    end if
end for
for each $k \in C2$ do
    $N_k \leftarrow$ neighboring frames of $k$
    $a \leftarrow$ sum of L1_norm($f$, $k'$) over $k'$ in $N_k$
    $k_{best} \leftarrow k'$ in $N_k$ maximizing L1_norm($f$, $k'$)
    add $\langle k_{best}, a\rangle$ to $C3$
end for
sort $C3$ by $a$
for $\langle k, b\rangle \in C3$ do
    if $b > \beta\cdot$(max $a$ in $C3$) then
        add $k$ to $Candidates$
    end if
end for
return $Candidates$
```

$$C_f^1 := \left\{ \begin{matrix} \langle k, n\rangle \mid k \in KF,\, L_f \cap L_k \neq \emptyset \\ n = |L_f \cap L_k| \end{matrix} \right\} \tag{3}$$

$$C_f^2 := \left\{ k \mid \langle k, m\rangle \in C_f^1,\, m > \alpha \cdot \underset{\langle k,n\rangle \in C_f^1}{max}(n) \right\} \tag{4}$$

$$C_f^3 := \left\{ \begin{matrix} \langle k, a\rangle \mid k_c \in C_f^2,\, k = \underset{k_c{}'}{argmax}(\|\mathbb{L}_f, \mathbb{L}_{k_c{}'}\|_1) \\ a = \sum_{k_c{}'} \|\mathbb{L}_f, \mathbb{L}_{k_c{}'}\|_1 \end{matrix} \right\} \tag{5}$$

Here, KF is a set of all keyframes in the map, $k_c{}'$ is a neighboring keyframe of k_c in C_f^2, and a is the summation of L1-norm of augmented scene representation for all keyframes in $k_c{}'$. Finally, keyframe candidates C_f are selected as in equation 6. This equation selects a keyframe with constraints such that its sum of L1-norms b is greater than the maximum b of C_f^3 multiplied by constant $\beta < 1$.

$$C_f := \left\{ k \mid \langle k, b\rangle \in C_f^3,\, b > \beta \cdot \underset{\langle k,a\rangle \in C_f^3}{max}(a) \right\} \tag{6}$$

3. Experiments

We analyzed the performance of our proposed framework and showed its operations in SLAM system on 3D photo-realistic environments and the real-world. In each of these operations, we evaluate the position errors during mapping and success rates of the relocalization task with its distance errors of the estimated positions.

Since we focused on using semantic labels to augment binary descriptors and the bag-of-words representation based on geometric features, the experimental results are compared to the RGB-D version of ORB-SLAM system (ORB-SLAM2) [15, 16]. Although there have been many deep neural network-based methods recently [25, 1, 14, 13], ORB-SLAM2 is still an advanced method that works robustly in real-world environments based on advantages of the ORB descriptor.

3.1. Experimental Setup

First, we tested our framework on the ScanNet [4] which provides large-scale indoor RGB-D scans consisting of 1513 reconstruction trajectories taken from 706 different environments, with 2.5M frames in total, along with dense 3D semantic annotations obtained manually via Mechanical Turk. The raw data was recorded from a structured light depth camera which returns absolute depth values.

We used RGB and depth image sequences as input frames for SLAM, and the annotated labels were used to extract semantic features in a pixel-wise manner. To build an evaluation dataset for the relocalization test, we extracted data from every 30th frame in each environment of the dataset. Maps for each of 201 environments in the ScanNet dataset were then created using ORB-SLAM2 and the SLAM based on our framework, respectively.

Our mapping procedure is designed to rewind 50 frames and to restart the procedure again when tracking-failure happens. When it continues to fail after five restarts, the system removes the problematic frame, rewinds 50 frames, and resets. This process is essential because the tracking fails frequently during the mapping phase, which causes map deficiency in several frames.

3.2. Evaluation Procedure

To simulate tracking-failure situations, our evaluation system displays a blank frame between different test frames. All test frames are repeated five times for relocalization process, and returns its camera position if it succeeds. The success rate of relocalization tasks is the frequency of successful cases in which the camera position is returned at least once in five repeated test frames. We use an estimated ground-truth based on the camera poses with neighboring frames for the evaluation of relocalization tasks, because the original poses in the dataset are not actually corresponding to our test frames in simulated tracking-failure status.

To avoid any accidental results and noises in experiments, we use the median value derived from 10 trials.

3.3. Performance

Table 1 presents the quantitative experimental results consisting of the absolute and relative mapping errors, success rate of the relocalization task, and the distance between the estimated position and the ground-truth position measured in different environments.

In the table, the higher success rate of relocalization indicates that the SLAM system was more successful at escaping the simulated tracking-failure status, and the lower average distance error indicates that the estimated poses after the relocalization are more accurate.

While the mapping qualities of both systems have no big difference, our system showed superior performances for the success rates of relocalization tasks (Total Avg.=78.95%) at the most of the environments compared to the ORB-SLAM2 system (Total Avg.=71.17%), which means that our framework enabled the system more robust and reliable.

Figures 2b and 2c present relocalization task with trajectories tested on one of the office environments in the dataset. The results show that relocalization success rate of our system (77.78%) is greater than ORB-SLAM2 (66.67%) with less distance error while both systems drew similar trajectories to the ground-truth shown in Figure 2a.

3.3.1 Effects Related to the Number of Objects

To investigate the relocalization performance with respect to the number of objects observed in each environment, we analyzed the experimental results by partitioning them according to the number of objects in each sample of the dataset. Considering the fact that even the environments with the same type may contain different numbers of objects, we categorized environments according to the ratio of the number of objects to the size of the environment.

Figure 3 presents the relocalization success rates, averaged for each category grouped by the number of objects corresponding 0 to 4, 4 to 8, 8 to 16, and 16 or more, respectively.

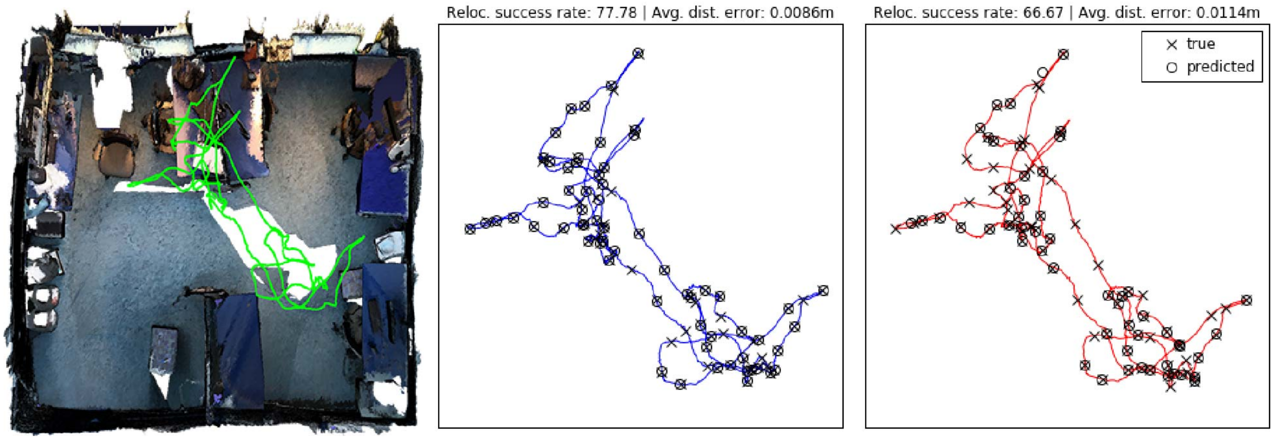
The experimental results show that the success rates of our system are greater than ORB-SLAM2, and we also discovered that the performance slightly increases according to the number of objects. These findings indicate that our proposed module works well for enhancing spatial abilities through the object-aware augmentation as we intended.

3.3.2 Robustness over Trajectory Length

To examine robustness over trajectory length, we analyzed the change in absolute position errors as the mapping progressed. Figure 4 presents the average of absolute position

Environments	Num. of Scenes	Absolute Position Error (m)		Relative Position Error (m)		Relocalization Success Rate		Mean Distance (m)	
		Ours	ORB-SLAM2	Ours	ORB-SLAM2	Ours	ORB-SLAM2	Ours	ORB-SLAM2
Bathroom	27	0.0929	0.0898	0.0110	0.0117	**0.7534**	0.6918	0.0277	0.0166
Bedroom	25	0.1419	0.1455	0.0124	0.0129	**0.7985**	0.6607	0.0357	0.0283
Bookstore	10	0.2594	0.2437	0.0114	0.0114	**0.8381**	0.7276	0.0104	0.0095
Classroom	7	0.2077	0.2300	0.0134	0.0132	**0.8717**	0.8248	0.0133	0.0360
Conference Room	13	0.6619	0.6808	0.0139	0.0147	**0.7192**	0.6441	0.0388	0.0727
Copy/Mail Room	7	0.4954	0.4799	0.0164	0.0163	0.7028	**0.8159**	0.0161	0.0186
Hallway	8	0.1149	0.1238	0.0126	0.0129	**0.8417**	0.8076	0.0125	0.0150
Kitchen	16	0.2055	0.2112	0.0103	0.0108	**0.8251**	0.7461	0.0183	0.0159
Living room	34	0.1561	0.1479	0.0132	0.0130	**0.7580**	0.6418	0.0294	0.0257
Lobby	8	0.2228	0.2153	0.0143	0.0132	**0.8274**	0.5860	0.0162	0.0153
Office	22	0.1360	0.1457	0.0104	0.0105	**0.8486**	0.7778	0.0117	0.0253
Misc.	24	0.1510	0.1457	0.0112	0.0111	0.7766	**0.7791**	0.0126	0.0234
Total	201	0.1994	0.1997	0.0121	0.0123	**0.7895**	0.7117	0.0226	0.0252

Table 1: Experimental results on global mapping and relocalization tasks. Small sampled environments are grouped as Misc.



(a) Ground-truth of a trajectory

(b) Ours

(c) ORB-SLAM2

Figure 2: Evaluation results on an office scene. (a) A trajectory of a hand-held camera on a 3D reconstructed office. (b) Evaluation results of the relocalization on the map created using our framework and (c) the results using ORB-SLAM2. The markers 'x' and 'o' stand for the actual location and the estimated location, respectively, thus the overlapped ones indicate that the relocalization has been successful.

errors of all evaluated environments, according to the number of frames processed. Our system and ORB-SLAM2 show similar error curves at the early stage, but our system presents lower errors when more frames are observed. The reason for the drastic decrease in the error at about 1300th frame is due to a statistical problem caused by rapid shortage of the number of scenes that have 1300 or more frames.

3.3.3 Efficiency of the Keyframe Candidate Selection

To evaluate the efficiency of the proposed algorithm for keyframe candidate selection, we investigated that declines in the number of keyframes in comparison with the number of the initial keyframes detected in the observed scenes. Typically, only a single keyframe (66.18%, average 1.89) remained after the whole selection process performed on initial keyframes. The number of keyframes decreased to 99.36%, 2.18%, and finally, 0.93% from the total number of keyframes were detected in each scene, throughout three steps of the algorithm.

3.4. Evaluation on Real Environment

We demonstrated and evaluated the efficacy of the proposed framework in a real environment by using a mobile robot platform, for which we computed the success rate of relocalization and the distance error.

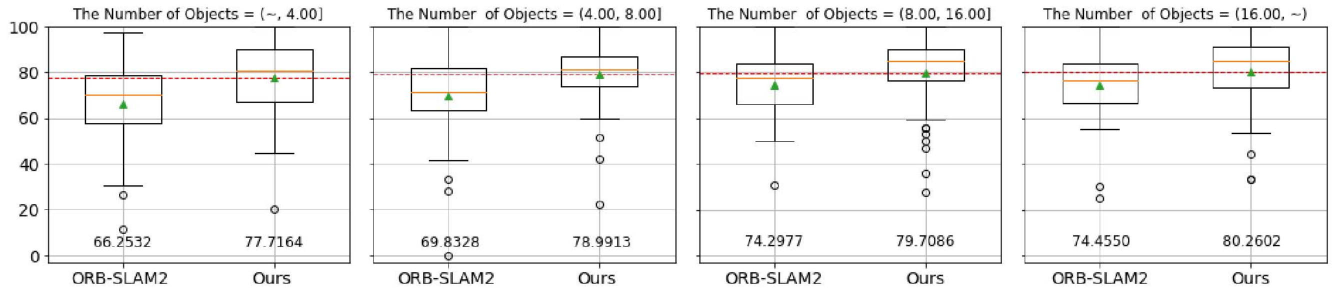


Figure 3: Relocalization performance considering the number of objects in the environments. Orange lines and green triangles in each box indicate the median and mean values, respectively, and red dashed lines stand for the mean value of our system.

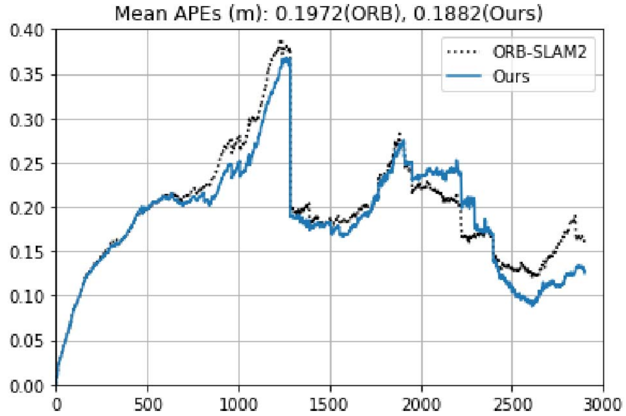


Figure 4: Absolute position errors along the increase of observed frame

3.4.1 Experimental Setup

To test the performance of our framework running in real-world scenarios, we built a model house. The house takes up 60 square meters and consists of four different spaces, including a living room, a kitchen, a bedroom and an entrance area as shown in Figure 5b.

We combined a map created using our visual SLAM system into a 2D cost map of the ROS navigation stack. Objects placed in the environment are annotated using a CNN-based object detector. From RGB data, the object detector extracts bounding boxes and class labels for each object in the observed scene. We used a YOLOv3 [18, 19] model trained on the MS-COCO dataset [12] in consideration of overall performance and computational cost. The map created based on our framework is depicted in Figure 5a.

The computational cost depends on the object classification model. In our experimental settings, the proposed framework uses 1.6 GB of GPU memory and the implemented SLAM system runs at 30 fps. This real-time performance allowed us to apply visual SLAM in mobile robots for real-world operations.

3.4.2 Robot Platform

We used a Pepper of Softbank Robotics, a child-sized mobile robot, for real-world tests. The robot has an omnidirectional drivetrain and two 5-DOF arms that can be used for simple object manipulation and gesture-based human-robot interaction. It has several sensors including a 3D sensor (ASUS Xtion), two RGB cameras (OmniVision OV5640), a four-microphone array, three laser range sensors, two infrared sensors, and two sonars. We recorded video frames using a RGB-D camera, which returns absolute depth values.

3.4.3 Results

Figures 5c and 5d show trajectories of the robot during mapping processes and the results of relocalization tasks evaluated on the model house. Although many parts of the house including walls, floors, the dining table and chairs are made of similar wooden patterns, which easily can cause tracking-failure, both systems estimated reasonable poses concerning to the structure of house.

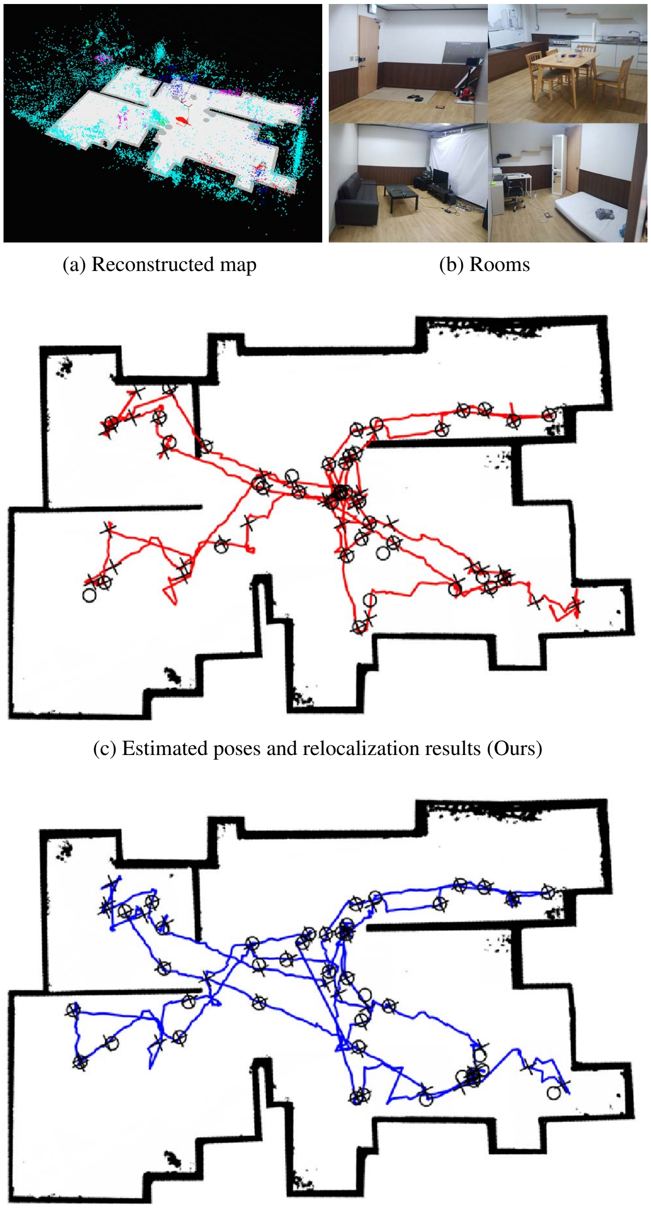
Relocalization success rates and accuracy measured for each scene of the dataset are presented in Table 2. We found that the robot recovered from the tracking-failure status more successfully using our system, and the accuracy of estimated pose after relocalization is also significantly higher for our system compared to the ORB-SLAM2 system.

	Reloc. success rate (%)	Average distance (m)
Ours	79.25	0.0512
ORB-SLAM2	71.43	0.0945

Table 2: Relocalization success rate and its accuracy of robot experiment compared to ORB-SLAM2.

4. Discussion

In Section 2.1, we defined γ as the constant factor to balance the weight of the semantic-augmented visual vocabu-



(a) Reconstructed map

(b) Rooms

(c) Estimated poses and relocalization results (Ours)

(d) Estimated poses and relocalization results (ORB-SLAM2)

Figure 5: Evaluation results on real environment. (a) Generated 2D occupancy grid map and 3D map points with semantic features. (b) Four different areas in our experimental model house. (c) The relocalization procedure on 2D occupancy grid map created based on our framework and (d) ORB-SLAM2. The markers x and o stand for the actual location and the estimated location, respectively, thus the overlapped ones indicate that the relocalization has been successful.

lary by determining the weight between semantic and visual vocabulary. We have empirically found that the semantic feature is appeared to dominate the visual feature when γ is over 1.0. In our experiments, we assumed that semantic features extracted by the object detection is reliable, so thus γ has been set to 1.0.

Unless enough semantic information is given, our framework depends more on the visual information that consists of geometric features. For this reason, the performance of our framework hardly goes below that of ORB-SLAM2, even for feature-sparse environments such as urban navigation scenarios.

We determined that environments with little semantic information is not specialized to our proposed framework and therefore focused on large-scale indoor environments containing many types of objects that can be observable in daily life.

Finally, we argue that the proposed framework applies not only to a specific SLAM system but to any conventional visual SLAM system in a generic plug-in way. As mentioned above, this plug-in method has an advantage that its performance can be improved by using semantic features while keeping the performance of visual SLAM at the base performance.

5. Conclusion

In this paper, we have presented a spatial perception framework that uses an object-aware visual scene representation.

The proposed framework is implemented into a visual SLAM system as an add-on feature augmentation module that fuses geometric features with corresponding semantic features, and effectively enhanced the performance of several vital processes including feature matching and relocalization. To this end, we also proposed an improved similarity measure of descriptors and an algorithm that enables to select keyframe candidates more efficiently working in our framework.

The proposed framework has been tested and evaluated using a mobile robot platform to validate its operation in the real environment, and a large-scale 3D photo-realistic dataset composed of several indoor environmental scenes.

Experimental results showed that our framework-based spatial perception system has relatively higher mapping performance when compared to ORB-SLAM2, and successfully executed relocalization tasks with superior performances.

Additionally, we analyzed the experimental results with respect to the number of objects observed in the environment, and discovered that our system's performance is increased according to the number of objects, which indicates the efficacy of the object-aware augmentation. An analysis on trajectory length proved that our framework takes better performance when it has observed the sufficient number of objects as the mapping proceeds.

Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-17-1-4128 and partly supported by the ICT R&D program of MSIT/IITP (2015-0-00310-SW.StarLab, 2017-0-01772-VTT, 2018-0-00622-HandAI, 2019-0-01367-BabyMind, 10060086-RISF, 2017-0-00162-Human-care Robot).

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 4

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. 1

[3] S. Choudhary, A. J. Trevor, H. I. Christensen, and F. Dellaert. Slam with object discovery, modeling and mapping. In *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1018–1025, 2014. 2

[4] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 10, 2017. 4

[5] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV)*, page 1403, 2003. 1

[6] A. J. Davison. Futuremapping: The computational structure of spatial ai systems. *arXiv preprint arXiv:1803.11288*, 2018. 1

[7] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 6:1052–1067, 2007. 1

[8] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006. 1

[9] J. H. Fecteau and D. P. Munoz. Salience, relevance, and firing: a priority map for target selection. *Trends in Cognitive Sciences*, 10(8):382–390, 2006. 1

[10] J. J. Gibson. *The perception of the visual world.* Houghton Mifflin, 1950. 1

[11] J. P. Gottlieb, M. Kusunoki, and M. E. Goldberg. The representation of visual salience in monkey parietal cortex. *Nature*, 391(6666):481, 1998. 1

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 6

[13] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2015. 4

[14] N. Merrill and G. Huang. Lightweight unsupervised deep loop closure. *arXiv preprint arXiv:1805.07703*, 2018. 4

[15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 4

[16] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 4

[17] Z. W. Pylyshyn. *Seeing and visualizing: It's not what you think.* MIT Press, 2003. 1

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 6

[19] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 6

[20] 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 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359, 2013. 2

[21] D. Scaramuzza and F. Fraundorfer. Visual odometry. *IEEE Robotics & Automation Magazine*, 18(4):80–92, 2011. 1

[22] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1

[23] R. Shadmehr and S. Mussa-Ivaldi. *Biological learning and control: how the brain builds representations, predicts events, and makes decisions.* MIT Press, 2012. 1

[24] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid. Meaningful maps with object-oriented semantic mapping. In *Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5079–5085, 2017. 2

[25] K. Tateno, F. Tombari, I. Laina, and N. Navab. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6243–6252, 2017. 2, 4

[26] B. Williams, G. Klein, and I. Reid. Real-time slam relocalisation. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007. 1

[27] B. Williams, G. Klein, and I. Reid. Automatic relocalization and loop closing for real-time monocular slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1699–1712, 2011. 3