Local to Global: Efficient Visual Localization for a Monocular Camera

Sang Jun Lee[1,2], Deokhwa Kim[1], Sung Soo Hwang[2], Donghwan Lee[1]
[1]NAVERLABS, Korea
[2]Handong Global University, Korea
{eowjd4}@naver.com, {deokhwa.kim, donghwan.lee}@naverlabs.com, {sshwang}@handong.edu

Abstract

Robust and accurate visual localization is one of the most fundamental elements in various technologies, such as autonomous driving and augmented reality. While recent visual localization algorithms demonstrate promising results in terms of accuracy and robustness, the associated high computational cost requires running these algorithms on server-sides rather than client devices. This paper proposes a real time monocular visual localization system that combines client-side visual odometry with server-side visual localization functionality. In particular, the proposed system utilizes handcrafted features for real time visual odometry while adopting learned features for robust visual localization. To link the two components, the proposed system employs a map alignment mechanism that transforms the local coordinates obtained using visual odometry to global coordinates. The system achieves comparable accuracy to that of the state-of-the-art structure-based methods and end-to-end methods for the visual localization on both indoor and outdoor datasets while operating in real time.

1. Introduction

The interest in visual localization (VL) has increased due to its key role in the visual navigation systems of autonomous cars, robots, and augmented reality (AR) applications. Recent VL algorithms have achieved centimeter-level accuracy for small room-size scenes and sub-meter-level accuracy in city-scale environments [5, 30, 39]. However, the majority of these algorithms measure the camera pose of each input image independently, even for sequentially correlated images, such as video frames. The lack of the consideration of the sequential correlation among images results in poor spatial-temporal consistency [13, 29] and non-smooth, or even jittered, camera poses, leading to the significant degradation of the user experience of real-world services, such as AR navigations [1, 2].

Many different deep learning-based VL methods have been proposed recently [6, 14, 39, 42]. Among them, structure-based methods, such as SuperPoint [9], R2D2 [28], and D2Net [10], use learned features to enhance local feature matching, and therefore, increase the robustness of localization. On the other hand, pose-regression-based methods use deep learning to approximate highly non-linear functions mapping an input image to camera poses [16, 17, 18]. There are also VL algorithms that use sequential images as input [6, 14, 39, 42]. However, the high computational cost of the existing learning-based VL methods requires running them on servers rather than client devices, such as mobile robots or smartphones. Moreover, some of the deep learning-based methods such as PoseNet [18], require time-consuming pre-training of models before applying them to new scenes or locations. The lack of the generalization ability of theses models makes them inapplicable to large-scale localization services [31].

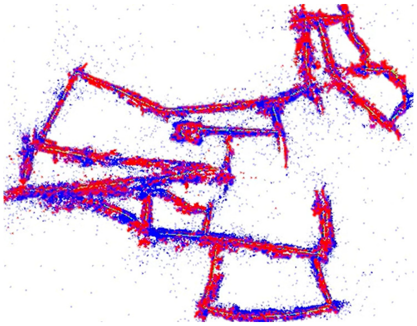


Figure 1. Visual localization of the proposed system employing only one monocular camera. The blue dots represent offline map points, while the red dots represent map points generated from an online map. The proposed system aligns the *local* online map to the *global* offline map. The green line shows an estimated trajectory on the online map.

This paper proposes a real time monocular visual localization system comprising two sub-systems responsible for client-side visual odometry (VO) and server-side VL, respectively. The VO sub-system generates local maps and estimates the relative movements of image frames [25]. The VO sub-systems can operate about 30 frames-per-second (fps) on a laptop without any graphics processing units

Method	Heterogeneous feature	Sensor	Visual odometry	Scale estimation	Place recognition
[4]	× (SIFT')	Mono	×	×	GPS/WiFi/user
[15]	× (SIFT')	Mono+IMU	✓	IMU	Mono
[22]	× (CONGAS)	Mono+IMU	✓	IMU	GPS/WiFi
[24]	✓(BRISK+SIFT)	Mono+IMU+GPS	✓	IMU	GPS
[40]	× (SIFT)	Mono+GPS	✓	Mono	GPS
Ours	✓(ORB+SuperPoint)	Mono	✓	Mono	Mono

Table 1. Comparison of various methods based on the server and client approach for structure-based sequential localization.

(GPUs) owing to the use of handcrafted features only. In the VL sub-system, this paper adopts a hierarchical visual localization method similar to [29]. This method utilizes learned features to perform robust structure-based localization, while adding the sequential information of consecutive image frames generated by a probabilistic place recognition module. To bridge the gap between the heterogeneous features of the two sub-systems, this paper proposes a new structure, called SuperKeyFrame, to process the heterogeneous features for the overall process of the proposed system. This approach reduces the computational cost of the server-side VL sub-system, by reducing the repetition of recalling the learned features.

Furthermore, this paper presents a map alignment (MA) sub-system to transform the local maps generated by the VO sub-system to the globally localized positions obtained from the VL sub-system. The proposed system uses the Kalman Filter to estimate a scale factor, which is necessary for a monocular camera-based system. To refine the robustness of the scale factor estimation, this paper re-trains the learned features by guiding the positions of the handcrafted features to improve the spatial consistency among all the heterogeneous features. Figure 1 shows the localized online maps on the global coordinates by using the proposed system.

The proposed system is demonstrated to perform on par with state-of-the-art structure-based VL methods while operating in real time performance. Such promising results establish the system's real time applicability to embedded devices in estimating monocular camera poses on global coordinates, and suggest the system's appropriateness for real-world VL applications.

The rest of this paper is organized as follows. Related work is discussed in Section 2, while the proposed system is presented in Section 3. The experimental setup and results are reported in Section 4. Section 5 concludes the paper.

2. Related Work

2.1. Structure-based visual localization

Structure-based VL methods compute camera poses using the Perspective-n-Point (PnP) solver with RANSAC [20]. In many cases, 3D-2D correspondences, called 3D-2D data association, between the query image and a given 3D model must be found to define the PnP problem. The 3D structures, which consist of key points with global 3D coordinates and their descriptors, are often built using visual simultaneous localization and mapping (SLAM) [11, 12, 19, 25, 26] or the structure from motion (SfM) algorithm [7, 32]. Active Search [30] was proposed for prioritized matching method to find accurate corresponding features. HF-Net [29] shows the state-of-the-art performance on the localization tasks by using learned features, NetVLAD [19] and SuperPoint [9] for the robust image retrieval and feature matching, respectively. Structure-based methods usually achieve better accuracy compared to learning-based end-to-end methods. However, their procedure of feature extraction is computationally expensive [42]. Furthermore, both approaches suffer from frequent jitters and inaccurate poses in image sequences since they have originally been proposed for single-image localization.

Existing methods for structure-based localization using image sequences use the server-client architecture, where the VL pipeline is executed on the server-side, while the VO pipeline is executed on the client-side [4, 15, 22, 24, 40]. The majority of these methods utilize sensors such as inertial measurement units (IMUs) or the global positioning system (GPS) to solve the complexity of place recognition and scale ambiguity.

This paper presents a low-cost system that requires a monocular camera without any extra sensors. Since a monocular camera cannot easily infer the scale factor without using IMUs or GPS, the novelty of the proposed system is the employment of a robust scale factor estimator utilizing the heterogeneous features. Table 1 highlights the differences between the proposed system and other structure-based localization methods.

2.2. Learning-based visual localization

Learning-based VL methods infer camera poses using on deep neural networks. PoseNet and its variations directly regress 6-DoF camera poses using CNNs [16, 17, 18] in an end-to-end fashion. DSAC [5] incorporates a deep learning model into the RANSAC scheme by adapting learnable parameters. However, as discussed in [31], data-driven learning methods lack the generalization ability and thus perform poorly in unseen locations.

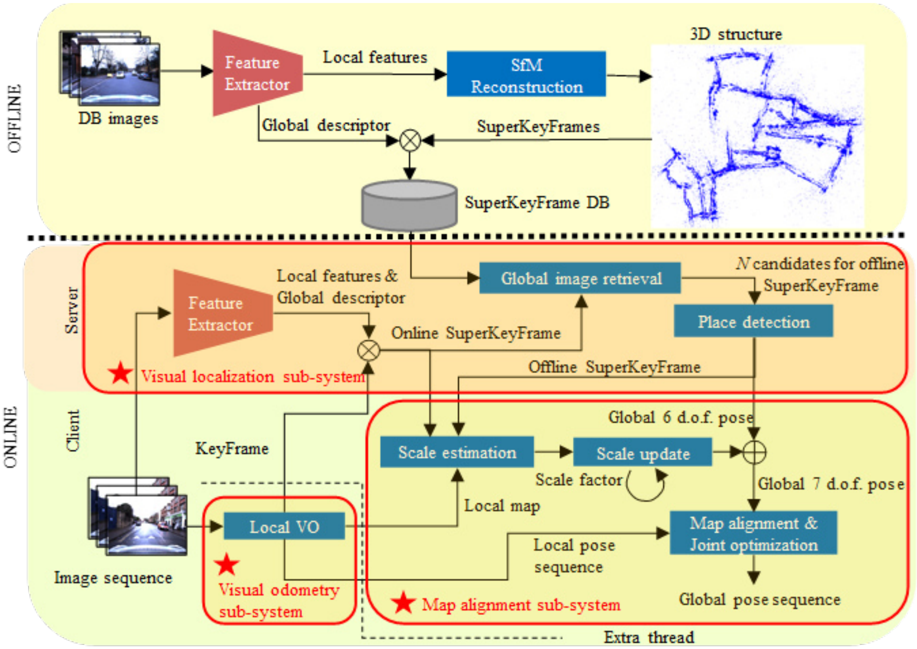


Figure 2. Overview of the proposed system.

Among the learning-based methods, the methods using image sequences are proposed to enhance the performance of localization [6, 14, 39, 42]. For example, vLocNet [39] employs consecutive two images to predict global poses and relative poses between them. Map-Net [6] learns geometric constraints using a 3D structure with odometry information obtained from a visual SLAM system. However, learning-based VL methods are typically less accurate than the structure-based methods and have many limitations as discussed in [31].

3. Method

3.1. System overview

Figure 2 shows the overview of the proposed system. In the offline phase, global and local features of database images are extracted in the feature extractor module using the SuperPoint method [9], that learns local features. Global image descriptors are extracted using the NetVLAD architecture [3]. Then, the 3D structure is reconstructed by using the SfM algorithm using the extracted local features as a offline map, and all the reconstructed views in the 3D structure become SuperKeyFrames. As a result, each SuperKeyFrame contains a global image descriptor, learned local features that correspond to 3D points in the 3D structure, and the camera pose of the frame.

In the online phase, the proposed system performs server and client-side operations. On the server-side, the structure-based VL sub-system learns features using a model running on GPUs. The proposed VL algorithm is similar to the HF-Net [29], except that it utilizes not only the global descriptor of the current input image, but also the place recognition results of the previous input images. On the client-side, the VO sub-system process sequential images on the local coordinate. An extra thread is involved to run the MA sub-system responsible for aligning the local frames on the local map obtained by the VO sub-system to the global coordinates.

Some KeyFrames generated by the VO sub-system that contain information useful for MA from the local to global coordinates are selected as the SuperKeyFrames. The SuperKeyFrame comprises two different types of local features: features learned on the server-side, called SuperPoint features, and ORB features generated on the client-side. The SuperKeyFrame becomes the query of the VL pipeline; thus, its 6-DoF global pose and 3D key points are returned from the server. As soon as the client module receives responses from the server and finishes the MA process, the proposed system selects a next KeyFrame as the SuperKeyFrame.

A relative scale factor is required to align the local map generated by the VO sub-system with the global coordinates provided by the VL sub-system based on the 6-DoF global pose. This is because the VO sub-system generates the local map up to the scale. This relative scale factor is computed by utilizing both the learned and ORB features on the selected SuperKeyFrame and updated over time using the Kalman Filter. Finally, the obtained 7-DoF global pose can help align the local map with the global coordinates.

In the base of ORB-SLAM, the number of KeyFrames is less than the number of frames. Similarly, narrowing down SuperKeyFrames to fewer than KeyFrames helps improve

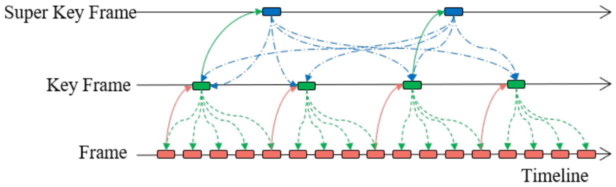


Figure 3. Timeline of the proposed system. The red, green, and blue boxes represent frames, KeyFrames, and SuperKeyFrames as the time sequence, respectively. Each solid line denotes the dependency of inheritance in the arrow direction, while each dashed line indicates that a started node provides estimated poses in the arrow direction.

the system performance by reducing the overhead of localization when using learned features. We illustrate the timeline of each kind of frame in Figure 3.

3.2. Data association

The proposed system requires two types of data association: 3D-2D (offline-online) data association used for 6-DoF VL, and 3D-3D (offline-online) data association mainly used for the scale factor estimation as described in Section 3.4.

First, 3D-2D correspondences are obtained using the following procedure. Given a pair of SuperKeyFrames between offline and online maps, 2D matching pairs of learned features are estimated using geometric verification [38]. Then, we can obtain the 3D points corresponding to the 2D points of the offline SuperKeyFrame from the 3D structure that has already been built. Further details on acquiring pairs of SuperKeyFrames are provided in Section 3.3.

3D-3D correspondences can be obtained by taking 3D points for the 2D points of the online SuperKeyFrame. However, these 3D points can only be acquired from the online map generated using VO. Since the VO sub-system creates the online map using handcrafted features, it requires the 2D-2D data association between the handcrafted features and learned features.

For the 2D-2D data association, the proposed system divides each SuperKeyFrame into multiple grid cells on the image plane and assigns ORB features to each of the closest grid cells. Then, based on the learned-feature position, it finds the closest ORB feature in grid cells within a certain radius. Using grid cells is more efficient than finding the closest ORB feature using brute-force algorithm.

3.3. Place recognition using sequential information

This section defines a probabilistic model for the place recognition using sequential information. The proposed model is similar to FABMAP [8] but is more suitable for scenarios with exceedingly sparse locations. Furthermore, it uses the NetVLAD model for image representation, rather than the BoW model [21].

Let $\mathcal{L} = \{L_1, L_2, ..., L_l\}$ denote a set of the discrete and disjoint l locations of SuperKeyFrames on the offline map. Each location in $\mathcal{L}$ is obtained from VL subsystem by solving the PnP problem [20] using the associated 3D-2D data. Since each SuperKeyFrame on the offline map has a NetVLAD descriptor, traditional image retrieval methods [27] searches the Top N candidate locations $\mathcal{C}^k = \{c_1^k, c_2^k, ..., c_N^k\} \in \mathcal{L}$ having similarity scores $S^k = \{s_1^k, s_2^k, ..., s_N^k\}$ given a query at time k. These scores are re-ranked using geometric verification [38].

The proposed model uses relative trajectories obtained between the VO and VL sub-modules to detect the true-positive places. This is because these relative trajectories should have the same tendency in terms of the temporal consistency when the places are true-positives. To this end, this paper assumes that the belief about the estimation of the current location depends on the previous location; the probabilistic model can then be represented by the recursive Bayesian estimation [37] as

$$p(\mathcal{C}^k|q^k, S^{1:k}) = \eta p(S^k|\mathcal{C}^k)p(\mathcal{C}^k|q^k, S^{1:k-1}), \quad (1)$$

where $p(\mathcal{C}^k|q^k, S^{1:k-1})$ is a prediction term for the belief probability, $p(S^k|\mathcal{C}^k)$ is a correction term, η is a normalization term, and q^k denotes the 6-DoF location with regard to the query at k; q^k can be obtained from the VO sub-system since it is consecutively tracked.

In the correction term, the normalized similarity scores can be obtained given $\mathcal{C}^k$ for the measurement update as

$$p(s_i \in \mathcal{S}^k|\mathcal{C}^k) = \omega \sum_{j \in |\mathcal{C}^k|} \rho(i,j)s_i \quad (2)$$

where ω is a normalization term divided by the sum of all elements in $\mathcal{S}^k$, while the function $\rho(i,j)$ returns one if i and j are the same to enforce the constraint that the measurement s_i is only given for the corresponding candidate c_j only.

The prediction term for the belief probability can be

$$\begin{aligned} &p(\mathcal{C}^k|q^k, S^{1:k-1}) \\ &\quad = \sum_{\mathcal{C}^{k-1}} p(\mathcal{C}^k|q^k, \mathcal{C}^{k-1})p(\mathcal{C}^{k-1}|q^{k-1}, S^{1:k-1}), \end{aligned} \quad (3)$$

expressed as

$$p(c_i \in \mathcal{C}^k|q^k, c_j \in \mathcal{C}^{k-1}) = \begin{cases} \gamma, & \nu(c_i, q^k, c_j) < \tau \\ 0, & else, \end{cases} \quad (4)$$

where γ is the ratio between the number of inlier features obtained from geometric verification and the number of all matching features, while $\nu(a,b,c)$ is a function that compares two relative transformations given the locations from

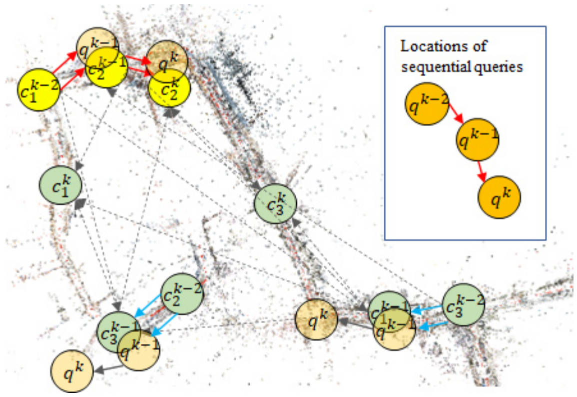


Figure 4. Place recognition using the proposed model. The circles represent the locations corresponding to SuperKeyFrames. The orange circles indicate the locations of consecutive queries. The yellow circles denote true-positive places in contrast to the green circles. The red arrows represent the trajectories that have the same tendency as the locations of the queries, while the blue arrows represent the false-positive trajectories. The black dashed arrows indicate wrong places that should be neglected.

a to b and a to c. The function determines whether both the differential angle and distance are under the threshold τ. Note that the motion model returns not the errors between relative trajectories but the image domain consistency between a query and a database image. This is because the latter is more important when the error of the relative pose is within the tolerated range, given that there can be different perspectives even at the same place.

Finally, a candidate location in $\mathcal{C}^k$ that maximizes the belief probability is detected as a true-positive location. Note that comparing relative trajectories in Equation (4) influences key impacts as shown in Fig 4. In the figure, the model for place recognition detects sequential candidate locations that follow the tendency temporally consist w.r.t. the locations of the sequential queries as true-positive locations.

The proposed place recognition model can be executed after the initial MA as described in Section 3.6. In other words, the initial place is detected by using only Equation (2), and then the model updates true-positive places using Equation (4). The place recognition module then sends the offline SuperKeyFrame according to the found places to the next processes.

3.4. Scale estimation

The proposed place recognition model detected an offline SuperKeyFrame according to the current online SuperKeyFrame in Section 3.3. 3D-3D correspondences between the two SuperKeyFrames are associated as described in Section 3.2. Then, the following poses transform such 3D point pairs, respectively; the 6-DoF poses estimated by the VL sub-system with regard to the offline SuperKeyFrame, the 6-DoF poses of the KeyFrames under the online SuperKeyFrames. In this way, the 3D points of the two SuperKeyFrames are exposed to the same features in the same place. Therefore, the scale factor can be calculated using the depth value of the 3D points.

The scale estimator computes the initial scale factor using the RANSAC scheme based on the Euclidean distance errors among scaled correspondences. Throughout the execution, it is updated over time using the Kalman Filter as suggested in [36]. Note that the scale factor is estimated relatively to the scale factor applied at the previous time point. To ensure that the scale factor is consistently updated, it is calculated after returning 3D points to the initial state. The more detailed formula for this process is provided in the supplemental document.

3.5. SuperORB

Finding a sufficient number of 3D-3D correspondences between offline and online maps is important for robust scale estimation. However, as the detectors for the SuperPoint and ORB features are different, precise correspondences cannot be found. To bridge the gap between two different local features in the VO and VL sub-systems, SuperORB is employed to produce points of common interest in the ORB detector for the same input image. The idea of SuperORB was inspired by SuperPoint [9], which performs homography adaptation considering different scaling. Similar to SuperPoint, SuperORB performs the homography adaptation process from the location of the extracted features. In other words, SuperORB is retrained considering different scaling while following the position of the ORB feature detector. Unlike SuperPoint that uses synthetic images for generating a pseudo-ground truth, SuperORB is trained to detect ORB features. The proposed keypoint detector and descriptor can be trained jointly.

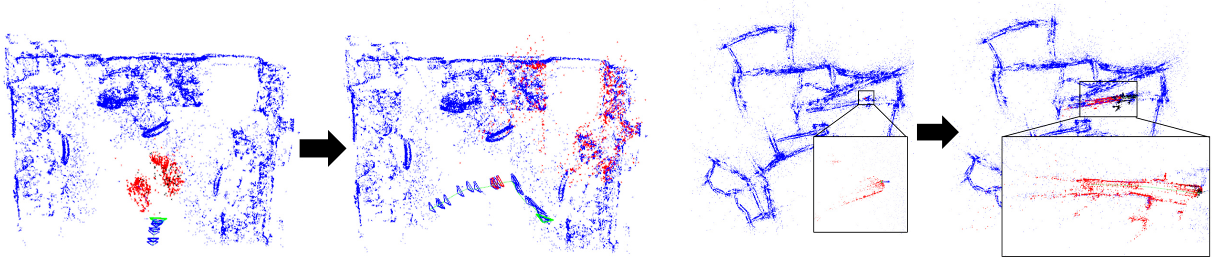
Table 2 lists the recall values obtgained for detecting local map, *i.e.*, ORB key points, when utilizing the SuperPoint and SuperORB detectors over the in 7-Scenes dataset [34]. Here, we define recall as the number of intersecting ORB features divided by the number of ORB features. Compared to the handcrafted ORB feature, SuperORB is more robust against the viewpoint and illumination changes because homographic and photometric data augmentation is applied as discussed in [25]. The supplemental document presents the evaluation of SuperORB.

	SuperPoint	SuperORB
Recall	0.651	**0.780**

Table 2. Recall of SuperPoint and SuperORB on the 7-scenes dataset [34].

3.6. Map alignment

So far, we demonstrated how to obtain the 6-DoF pose and the scale factor using the proposed modules. The mean



(a) 7-Scenes dataset.

(b) RobotCar dataset.

Figure 5. Examples of map alignment. Each left image shows the local map before map alignment, while each right image indicates the local map after the map alignment. Note that the blue points represent offline map points, while the red points represent online map points. The blue KeyFrames are obtained from visual odometry, the red Key Frames are obtained from visual localization, and the green Key Frames are the current Key Frames on visual odometry.

of the re-projection error from 3D-2D correspondences was used as a weight to compare the accuracy of the 6-DoF poses returned by the VO and VL sub-modules. This approach estimates a more smooth pose. The detailed expressions are provided in the supplemental document.

The MA sub-module calculates the relative 7-DoF similarity from a 6-DoF pose of current KeyFrame with the estimated scale factor obtained as the VO sub-system to the weighted 6-DoF pose with a scale set to 1 obtained as the VL sub-system. Finally, the similarity transformation aligns the local maps on the local coordinates to the global coordinates optimizing using 3D similarity constraints [35]. At this time, SuperKeyFrames are fixed to relax the accumulated errors in surrounding KeyFrames following the trajectory of SuperKeyFrames. Figure 5 shows an example of MA.

4. Experiment

Implementation details. The client-side of the proposed system was implemented in C++ using an i-9 CPU, and 32G RAM. All parameters were set as proposed in [25]. For the server-side, NetVLAD and SuperPoint architectures were implemented using the TensorFlow. Note that learning-based features were extracted using NVIDIA 1080 Ti GPUs. All parameters were set as suggested in [3, 9]. The remaining parameters of the proposed system was set as N=10, τ=1(m/°) for small-scale environments, and τ=10(m/°) for large-scale environments.

Datasets. To enable the evaluation, datasets are selected based on the following conditions: 1) a reference map is provided and 2) the dataset should be large enough to enable VO. The following two popular datasets were found to satisfy the conditions: 7-Scenes dataset [34], and Oxford RobotCar dataset [23]. The 7-Scenes dataset consist of indoor scenes in small-scale environments, whereas RobotCar consist of outdoor scenes in large-scale environments, which include a large amount of illumination and appearance changes. The reference maps by utilizing the provided depth information with poses for the 7-Scenes dataset and COLMAP [32] with GPS and inertial navigation system data provided for the RobotCar dataset.

Ablation study. The proposed system was tested from the following three perspectives: place recognition, scale estimation, and local feature generation. Two tests were conducted for place recognition, namely, with and without applying sequential information. For scale estimation, two different scale estimation methods were tested, namely, the proposed scale estimation method using the Kalman Filter and a method employing Dynamic Time Warping (DTW) proposed for cooperative SLAM [33]. Finally, SuperPoint and SuperORB were tested for local feature generation.

Comparison with other methods. The proposed system, which achieves the best performance in the ablation study, was compared to several existing structure-based and end-to-end methods. Note that we excepted comparison of other methods which use extra sensors such as IMU and GPS for fair comparisons.

4.1. Evaluation on the 7-Scenes dataset

Ablation study. Table 3 shows the results of the ablation study. The system employing sequential information demonstrated a better performance in place recognition than the system without sequential information. In addition, the system employing the proposed scale estimation method demonstrated a better performance than the one employing the DTW-based method, while the system employing SuperORB demonstrated a better performance than the one employing SuperPoint. Figure 6 shows the output trajectories of the version that achieves the best performance employing sequential information and the proposed scale estimation with SuperORB.

Comparison with other methods. Tables 4 and 5 list the results achieved by the proposed system and other considered methods. The proposed system outperformed the state-of-the-art vLocNet end-to-end method and achieved similar

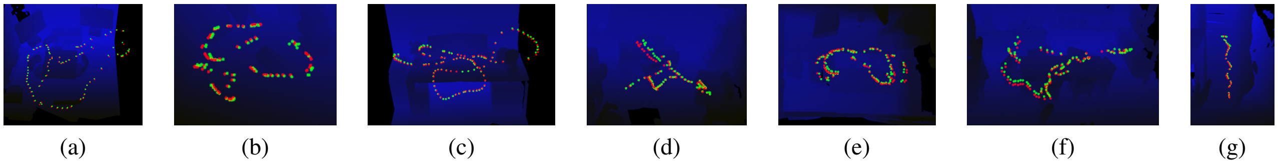

(a) (b) (c) (d) (e) (f) (g)

Figure 6. Trajectories output by the proposed system for the 7-Scenes dataset. The red dots indicate the ground truth camera positions of KeyFrames, while the green dots indicate the estimated camera positions of KeyFrames. (a) Chess, (b) Fire, (c) Heads, (d) Office, (e) Pumpkin, (f) Redkitchen, (g) Stairs.

P. R.	w/o seq.	w/ seq.		
S. E.	K. F.	DTW [33]	K. F.	
Feature	S. P.	S. P.	S. P.	S. O.
Chess	0.04/2.81	0.14/5.26	**0.02**/2.45	**0.02/2.34**
Fire	0.04/3.40	0.05/2.99	0.04/3.01	**0.02/2.25**
Heads	0.012/3.83	0.02/3.29	**0.01/2.51**	**0.01**/2.61
Office	0.05/3.40	0.16/6.18	**0.04/4.16**	**0.04/1.16**
Pump.	0.07/3.89	0.13/6.15	0.05/3.94	**0.04/3.92**
RedK.	0.05/5.05	0.24/25.83	0.05/4.49	**0.03/4.34**
Stairs	0.13/10.1	0.19/7.43	0.10/6.72	**0.03/2.93**

Table 3. Median translation and rotation errors for the ablation study on the 7-Scenes dataset (m/°). P.R., S.E., K.F., S.P., and S.O stand for Place Recognition, Scale Estimation, Kalman Filter, SuperPoint, and SuperORB, respectively.

	Structure-based method			Ours
	DSO [12]	A.S. [30]	DSAC++ [5]	
Chess	0.17/8.13	0.04/1.96	**0.02/0.5**	**0.02**/2.34
Fire	0.19/65.0	0.03/1.53	**0.02/0.9**	**0.02**/2.25
Heads	0.61/68.2	0.02/1.45	**0.01/0.8**	**0.01**/2.92
Office	1.51/16.8	0.09/3.61	**0.03/0.7**	0.04/1.16
Pump.	0.61/15.8	0.08/3.10	**0.04**/1.1	**0.04**/3.92
RedK.	0.23/10.9	0.07/3.37	0.04/1.1	**0.03**/4.34
Stairs	0.26/21.3	**0.03**/2.22	0.09/2.6	**0.03**/2.93

Table 4. Median translation and rotation errors of the proposed and existing structure-based methods on the 7-Scenes dataset (m/°).

	End-to-end method			Ours
	PoseNet [18]	MapNet [6]	vLocNet [39]	
Chess	0.13/4.48	0.08/3.25	0.04/**1.71**	**0.02**/2.34
Fire	0.27/11.30	0.27/11.69	0.04/5.33	**0.02/2.25**
Heads	0.17/13.00	0.18/13.25	0.05/6.65	**0.01/2.92**
Office	0.19/5.55	0.17/5.15	**0.04**/1.95	**0.04/1.16**
Pump.	0.26/4.75	0.22/4.02	**0.04/2.28**	**0.04**/3.92
RedK.	0.23/5.35	0.23/4.93	0.04/**2.21**	**0.03**/4.34
Stairs	0.35/12.40	0.30/12.08	0.10/6.48	**0.03/2.93**

Table 5. Median translation and rotation errors of the proposed system and end-to-end methods on the 7-Scenes dataset (m/°).

Pipeline	VL	VO	client-side (VO+MA)
Number of frames	1163	33665	33665
Frame per second(fps)	1.37	25	25
Accuracy(m/°)	4.887/**0.411**	N/A	**4.754**/0.585

Table 6. Decomposition of the proposed system and performance analysis for the Full sequence of RobotCar Dataset.

results to the state-of-the-art DSAC structure-based method. Note that DSAC is not suitable for real time VL as discussed in [5]. Furthermore, this method doesn't work well on large-scale scenes, as discussed in [41]. In contrast, the proposed system achieves robust and accurate localization in real time (25 fps) even on large-scale scenes, as discussed below.

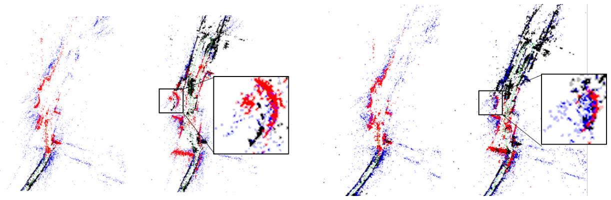
4.2. Evaluation on the RobotCar dataset

The Full and Loop sequences evaluated in [6] were used for the RobotCar dataset. The Full sequence comprises a long sequence of 9,562 m capturing a complex road environment. The Loop sequence is 1,120 m long and contains many eco-environments.

First, the performances of the VL and VO sub-systems were compared when the proposed method system was applied to the Full sequence of the RobotCar dataset. As shown in Table 6, the VL sub-system processed 1,163 SuperKeyFrames on the server-side, resulting in 1.37 fps. If the VL sub-system was used solely as a localization service for real-world applications, it would result in non-smooth camera movement because camera poses were obtained every 0.73 seconds. In contrast, the ORB-SLAM based VO sub-system processed 33,665 frames from the given 34K frames running on 25 fps on the client-side. The entire pipeline ran at up to 25 fps, achieving comparable accuracy to that of the VL sub-system. Hence, we argue that the client-side of the proposed system can be used for real-world applications because it operates in real time and provides camera poses using global coordinates.

Ablation study. Table 7 lists the results of the ablation study as the same setting for 7-Scenes dataset. For the indoor scenes, the system using place recognition with sequential information, Kalman Filter-based scale estimation, and SuperORB demonstrated the best performance.

SuperORB is more robust in the scale estimation. Figure 7 shows the result of scale estimation for the same place during performing a U-turn. In the case of using SuperORB, the map created before and after the U-turn coincides with the reference map, whereas in the case of using SuperPoint, there is a slight difference.



(a) (b)

Figure 7. Aligned map after a U-turn: (a) SuperPoint-based system and, (b) SuperORB-based system. The SuperORB-based system shows better map alignment due to more accurate scale estimation.

P. R.	w/o seq.	w/ seq.		
S. E.	K. F.	DTW [33]	K. F.	
Feature	S. P.	S. P.	S. P.	S. O.
Full seq.	9.04/1.52	15.17/1.78	5.52/1.01	**4.75/0.58**
Loop seq.	8.19/2.13	180.8/2.54	5.27/**1.97**	**5.23**/2.82

Table 7. Mean translation and rotation errors for the ablation study on the RobotCar dataset(m/°). P.R., S.E., K.F., S.P., and S.O. stand for Place Recognition, Scale Estimation, Kalman Filter, SuperPoint, and SuperORB, respectively.

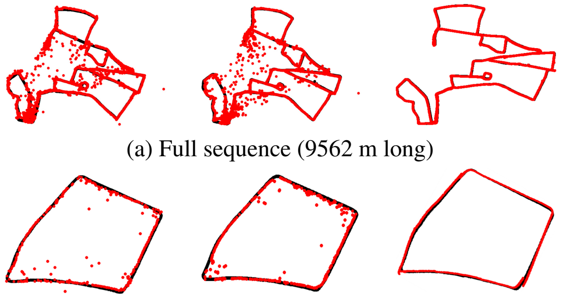
Comparison with other methods. Table 8 and 9 list the results achieved by the proposed system and other considered methods. It can be noticed from the tables that the proposed system outperformed the other methods. Among the structure-based methods, DSAC++ cannot be operated in outdoor scenes, while ORB-SLAM loses tracked poses. Note that all other methods use all frames for the VL pipeline, whereas the proposed system uses only 1K frames, as reported in Table 6. Some of the structure-based methods can perform only in non-real time with high computational overhead, whereas the proposed system can operate in real time with low computational overhead.

Furthermore, the proposed system outperformed the end-to-end methods with a large margin. The lowest performance of mean translation error among the other methods was over 13 m, whereas the proposed method proved a mean translation error of under 6 m. Figure 8 shows the results when applying the proposed and other methods.

The results of the experiment suggest that the proposed system performs successful VL in both small-scale and large-scale environments. The efficiency and accuracy of the system is especially impressive given that the experiment is done on a monocular camera system without any extra sensors.

5. Conclusion

This paper presented a real time monocular VL system that uses heterogeneous features obtained using client-side VO and a server-side probabilistic model for place recognition. To handle these heterogeneous features, the concept of SuperKeyFrame is introduced to link handcrafted and learned features while ensuring their spatial consistency. To align the coordinates between the two sub-systems, the system further employs a MA sub-system with a scale factor estimator that uses the heterogeneous features. According to the experimental results, the proposed system can achieve high efficiency with low computational cost and accuracy comparable to that of the state-of-the-art structure-based and end-to-end VL methods. The proposed system can be integrated into ORB-SLAM and any other SLAM/VO-based systems. Hence, we expect that the proposed system will be utilized in various industries as a useful low-cost localization module.



(a) Full sequence (9562 m long)

(b) Loop sequence (1120 m long)

Figure 8. Resulting trajectories on the RobotCar dataset for the proposed system and other methods: (a) Full sequence and (b) Loop sequence. In both (a) and (b), the first image is for PoseNet, the second image is for MapNet, and the third image is for the proposed system.

	Structure-based method			Ours
	DSAC++ [5]	ORB-SLAM [25]	Stereo VO [23]	
Full	N/A	N/A	80.32/13.73	**4.75/0.58**
Loop	N/A	N/A	22.42/45.50	**5.23/2.82**

Table 8. Mean translation and rotation errors for the proposed and structure-based methods on the RobotCar dataset (m/°).

	End-to-end method			Ours
	PoseNet [18]	MapNet [6]	AD-MapNet [14]	
Full	46.6/10.5	44.6/10.4	19.2/4.60	**4.75/0.58**
Loop	7.90/3.53	9.29/3.34	6.45/2.98	**5.23/2.82**

Table 9. Mean translation and rotation errors for the proposed and end-to-end methods on the RobotCar dataset (m/°).

Acknowledgement

This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2019-0-01309, Development of AI Technology for Guidance of a Mobile Robot to its Goal with Uncertain Maps in Indoor/Outdoor Environments)

References

[1] https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html.

[2] https://www.naverlabs.com/en/storydetail/112.

[3] R. Arandjelovic and et al. Netvlad: Cnn architecture for weakly supervised place recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] C. Arth and et al. Wide area localization on mobile phones. *In 2009 8th ieee international symposium on mixed and augmented reality*, 2009.

[5] E. Brachmann and et al. Dsac-differentiable ransac for camera localization. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017.

[6] S. Brahmbhatt and et al. Geometry-aware learning of maps for camera localization. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[7] Z. Cui and P. Tan. Global structure-from-motion by similarity averaging. *In Proceedings of the IEEE International Conference on Computer Vision*, 14(1):864–872, 2015.

[8] 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.

[9] D. DeTone and et al. Superpoint: Self-supervised interest point detection and description. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[10] M. Dusmanu and et al. D2-net: A trainable cnn for joint detection and description of local features. *CVPR*, 2019.

[11] J. Engel and et al. Lsd-slam: Large-scale direct monocular slam. *In Proc. of ECCV*, 14(1):834–849, 2014.

[12] J. Engel and et al. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2018.

[13] P. E.Sarlin and et al. Leveraging deep visual descriptors for hierarchical efficient localization. *In Conference on Robot Learning*, 2018.

[14] Z. Huang and et al. Prior guided dropout for robust visual localization in dynamic environments. *In Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[15] E. Jones and S. Soatto. Visual-inertial navigation, localization and mapping: A scalable real-time large-scale approach. *Intl. J. of Robotics Res*, 2011.

[16] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. *In ICRA*, 2016.

[17] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] A. Kendall and et al. Posenet: A convolutional network for real-time 6-dof camera relocalization. *In Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[19] S. J. Lee and S. S. Hwang. Elaborate monocular point and line slam with robust initialization. *In Proceedings of the IEEE International Conference on Computer Vision*, pages 1121–1129, 2019.

[20] V. Lepetit and et al. Epnp: An accurate o(n) solution to the pnp problem. *International journal of computer vision 81(2): 155*, 2009.

[21] S. Lowry and et al. Visual place recognition: A survey. *in IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[22] S. Lynen and et al. Large-scale, real-time visual-inertial localization revisited. *arXiv preprint arXiv:1907.00338*.

[23] W. Maddern and et al. 1 year, 1000km: The oxford robotcar dataset. *The International Journal of Robotics Research*.

[24] S. Middelberg and et al. Scalable 6-dof localization on mobile devices. *In European conference on computer vision*, 2014.

[25] R. Mur-Artal and et al. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[26] 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.

[27] J. Philbin and et al. Object retrieval with large vocabularies and fast spatial matching. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[28] J. Revaud and et al. R2d2: Reliable and repeatable detectors and descriptors for joint sparse keypoint detection and local feature extraction. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshop*, 2019.

[29] P. E. Sarlin and et al. From coarse to fine: Robust hierarchical localization at large scale. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.

[30] T. Sattler and et al. Efficient and effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016.

[31] T. Sattler and et al. Understanding the limitations of cnn-based absolute camera pose regression. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[32] J. L. Schonberger and F. Jan-Michael. Structure-from-motion revisited. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[33] P. Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855:1–23, 2008.

[34] J. Shotton and et al. Scene coordinate regression forests for camera relocalization in rgbd images. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[35] H. Strasdat and et al. Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2(3):7, 2010.

[36] E. Sucar and J.B. Hayet. Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift. *IEEE International Conference on Robotics and Automation*, 2018.

[37] S. Thrun and et al. Probabilistic robotics. *Communications of the ACM*, 45(3), 2002.

[38] H. Toepfer. Geometric verification. in: Jansen d. the electronic design automation handbook. *Springer, Boston, MA*, 2013.

[39] A. Valada and et al. Deep auxiliary learning for visual localization and odometry. *In 2018 IEEE International Conference on Robotics and Automation*, pages 6939–6946, 2018.

[40] J. Ventura and et al. Global localization from monocular slam on a mobile phone. *IEEE transactions on visualization and computer graphics*, 20(4):531–539, 2014.

[41] P. Weinzaepfel and et al. Visual localization by learning objects-of-interest dense match regression. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[42] F. Xue and et al. Local supports global: Deep camera relocalization with sequence enhancement. *In ICCV*, 2019.