Drone Systems and Applications

OPEN ACCESS | Article

Factor graph localization for mobile robots using Google Indoor Street View and CNN-based place recognition

Kusal B. Tennakoon[a], **Oscar De Silva**[a], **Awantha Jayasiri**[b], **George K.I. Mann**[a], **and Raymond G. Gosine**[a]

[a]Department of Mechanical Engineering, Memorial University of Newfoundland, St. John's, NL, Canada; [b]Aerospace Research Centre, National Research Council Canada, Ottawa, ON, Canada

Corresponding author: **Kusal B. Tennakoon** (email: kbtennakoon@mun.ca)

Abstract

This article proposes a mobile robot localization system developed using Google Indoor Street View (GISV) and Convolutional Neural Network (CNN)-based visual place recognition. The proposed localization system consists of two main modules. The first is a place recognition module based on GISV and a net Vector of Locally Aggregated Descriptors (VLAD)-based CNN. The second is a factor graph-based optimization module. In this work, we show that a CNN-based approach can be utilized to overcome the lack of visually distinct features in indoor environments and changes in images that can occur when using different cameras at different points in time for localization. The proposed CNN-based localization system is implemented using reference and query images obtained from two different sources (GISV and a camera attached to a mobile robot). It has been experimentally validated using a custom indoor dataset captured at the Memorial University of Newfoundland engineering building basement. The main results of this paper show that GISV-based place recognition reduces the percentage drift by 4% for the dataset and achieves a Root Mean Square Error (RMSE) of 2 m for position and 2.5° for orientation.

Key words: topological localization, indoor localization, factor graph, place recognition

1. Introduction

Mobile robots are used in many safety-critical applications involving human operators, and precise localization has become an essential requirement for the autonomy of a mobile robot. Recent advances in localization use Google Street View (GSV) to identify the current location of a robot in its environment. This procedure matches an image corresponding to the robot's current position with a database of images representing its environment using a process known as place recognition, which can bind the inherent drift in the odometers of robot platforms.

In literature, place recognition methods using GSV mainly use conventional feature-based methods or Convolutional Neural Network (CNN) feature-based methods. Conventional feature-based methods (Yu et al. 2017; Yan et al. 2018; Zhou et al. 2021) use conventional feature descriptors encoded as either a Bag of Words (BoW) or a set of Vector of Locally Aggregated Descriptors (VLAD). CNN feature-based methods (Arandjelovic et al. 2016; Maffra et al. 2018; Yin et al. 2019) use off-the-shelf CNNs customized as dense feature extractors. The extracted dense features are stored in a data structure compatible with efficient search. Some studies, such as the urban localization system proposed in Bresson et al. (2019), use the depth maps and the absolute positions of the locations in addition to the respective GSV panoramas. Several studies (Agarwal et al. 2015; Yu et al. 2016; Yan et al. 2018) have proposed metric localization methods by modelling the solution as a nonlinear least squares estimation problem solved using an optimization algorithm.

Google Indoor Street View (GISV) poses a challenge to these methods. GISV is not widely available compared with outdoor street view. Thus, there is a lack of indoor street view maps compared with outdoors. Second, indoor environments are abundant in repetitive and self-similar structures such as ceilings, flooring, and walls with visual similarity in buildings compared with outdoors. It can confuse the place recognition system ending up with incorrect predictions. Third, indoor environments lack robust and unique features making it challenging for conventional feature detectors to extract sufficient key points for image matching. Finally, frequent changes in an indoor environment can cause significant differences among reference and query images of the same location, making the matching process difficult.

In addition to the above challenges unique to indoor environments, GSV images, in general, are an equirectangular projection of the respective 360° images. These equirectangular images contain significant distortions to be handled by conventional feature descriptors. A study by Morel and Yu (2009) has shown how significant affine variance among images can impact the overall matching process.

The lack of indoor place recognition datasets has been addressed in literature by creating custom datasets to serve as reference images. One approach is constructing the reference dataset as an offline step separate from the online

Fig. 1. 360° image corresponding to node1 captured by Samsung Gear 360 camera.

localization process (Taira et al. 2021). The second approach is constructing the reference dataset online parallel to the localization process using a visual Simultaneous Localization And Mapping (SLAM) approach (Tateno et al. 2017; Maffra et al. 2018, 2019). Common methods used to penalize the effect of repetitive features and self-similar structures in these indoor scenes are burstiness weighting (Piasco et al. 2019) and term frequency-inverse document frequency (tf-idf) scoring (Kejriwal et al. 2016). Burstiness weighting is assessing the importance of a feature based on the number of times it repeats in an image. Tf-idf scoring prioritizes visual words that are less common within the entire dataset and penalizes those that are common among multiple images of the dataset. The lack of unique features and frequent changes in an environment are problems that are comparatively harder to tackle. However, studies (Tennakoon et al. 2021; Yu et al. 2019) have shown that CNN-based features are more robust and perform better than conventional handcrafted features in these cases. Studies have used two methods in handling the distortions present in equirectangular images. One method generates undistorted perspective views called virtual views from the equirectangular images (Majdik et al. 2013). The other method directly downloads perspective images of desired heading and tilt using GSV Application Programming Interface (API) (Agarwal et al. 2015; Kejriwal et al. 2016).

In this work, we have created our GISV dataset and uploaded it to the Google servers. To eliminate the effect of distortions, we download perspective images of desired heading and tilt angles corresponding to selected locations. These perspective images constitute the reference dataset. To minimize the effect of repetitive and self-similar structures, we select locations for the reference dataset such that the selected locations are visually distinct from one another based on the criteria that a human can easily distinguish the locations from one another. To overcome the lack of unique features and frequent changes in the environment, we exploit the benefits of CNN-based features by using a netVLAD CNN for the place recognition system.

Closely related to place recognition is a process termed loop closure used in mobile robot navigation. Popular studies using visual loop closure for robot navigation include Qin et al. (2018). Loop closure relies on a reference image database filled while traversing an environment using the onboard camera of a robot. As a result, a correct match between a reference and a query image can only be achieved when the robot revisits a previous location, i.e., during a loop closure. Once a match occurs, the drift of the robot is corrected in these studies using a constraint on its trajectory in a factor graph localization framework (Qin et al. 2018). Although conceptually similar, this drift correction step in loop closure does not transfer well to indoor street view-based localization. The loop closure procedure in Agarwal et al. (2015) and Qin et al. (2018) captures the reference and query images from the same camera. If the camera used is calibrated, the availability of the camera's intrinsic parameters is an advantage in estimating the relative pose between the query and the matching reference image. However, in the case of GISV place recognition, the reference images are perspective images downloaded using the GSV API. Hence, the intrinsic parameters of the camera used to capture the original street view images are no longer valid. The unavailability of depth information is a disadvantage for indoor street view-based localization. Even if the relative pose between a query image and a matching reference image is determined, the relative translation is only available up to a scale. It makes it more challenging to exploit all the available information for localization in a factor graph.

Fig. 2. The node map of the MUN Engineering building basement.

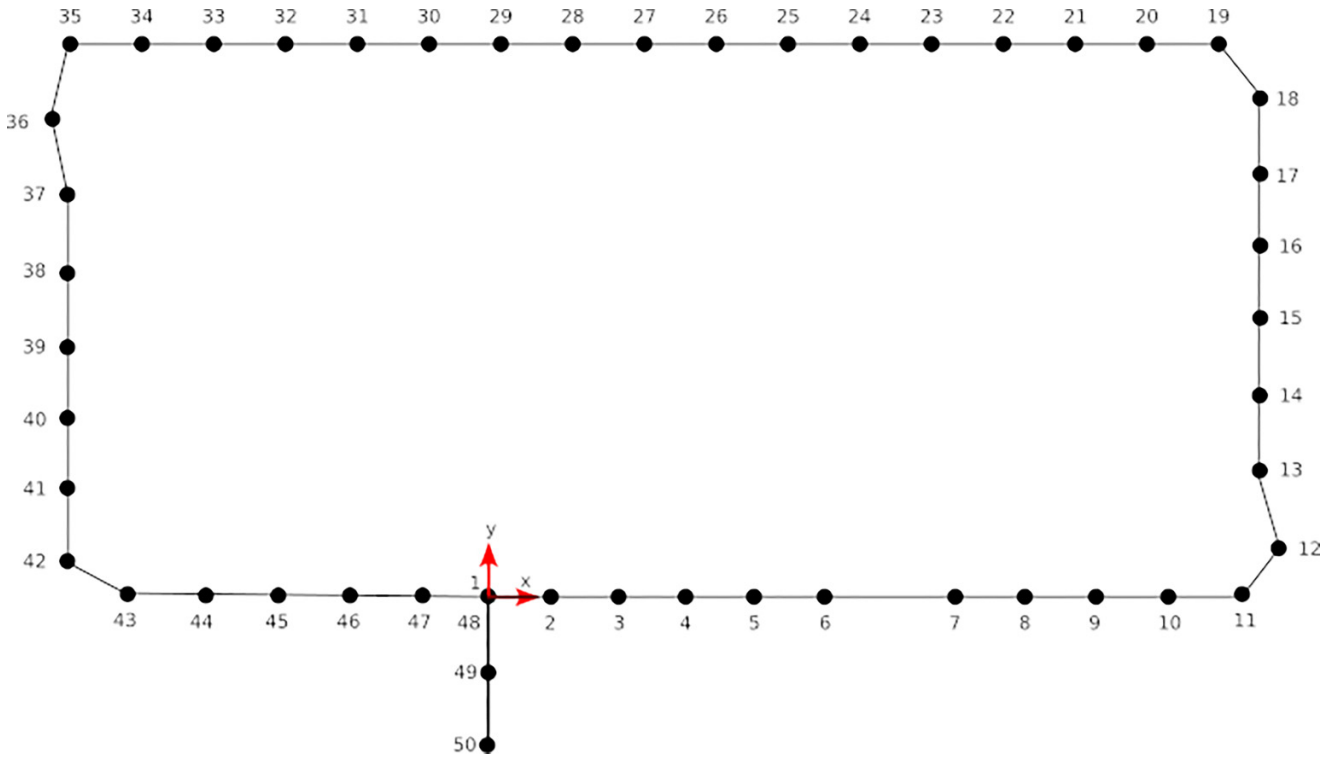


Table 1. Parameter values used in downloading perspective images from Street View images.

Parameter	Value
Image size	640 px × 640 px
Heading angles	0°, 60°, 120°, 180°, 240°, 300°
Pitch angles	0°
FOV	90°

In this study, we address the absence of intrinsic parameters for the reference images (perspective images downloaded from GISV) by deriving an equivalent intrinsic matrix using the metadata of GISV images. We address the issue of relative translation being up to a scale by assigning a scale factor with a representative noise associated with the matching performance of the CNN place recognition system. Combined, these make it possible to impose place recognition constraints in the factor graph localization framework to correct the odometry drift of the robot.

The contributions of the proposed indoor localization system are as follows.

1. Design of a GISV-based place recognition architecture suitable for indoor reference and query images obtained from two different cameras.
2. A method of incorporating GISV place recognition results in a robot localization factor graph.
3. Experimental validation of the place recognition module and the factor graph-based localization module using a custom dataset.

The paper is organized as follows. Section 1 introduces visual topological SLAM and the proposed system. Section 2 presents a detailed discussion on state of the art in visual topological localization. Section 3 provides an overview of the methodology followed in this study. Section 4 presents a discussion of the obtained results. Section 5 presents conclusions derived from this study. Finally, Section 6 discusses the future directives in the field of study.

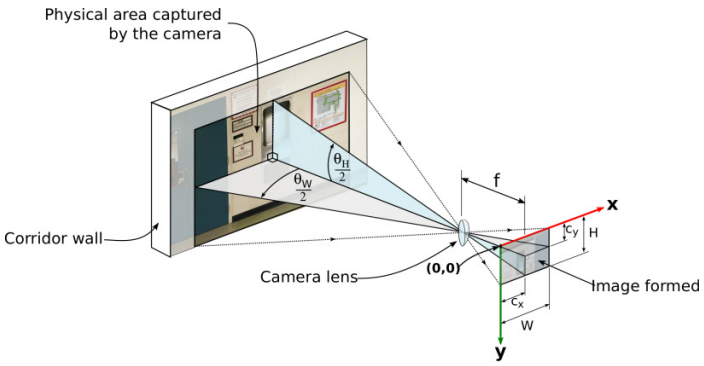
2. Related work

Vision-based topological localization and mapping methods employ visual place recognition as the mechanism for loop closure (Valgren and Lilienthal 2007; Ascani et al. 2008). A camera extracts useful visual information from the environment, and then compares it against a known set of information belonging to the same environment the robot navigates. Visual topological localization and mapping have appeared in different forms depending on the type of feature descriptor used, the detector–descriptor combination used, the search method used, and the type of third-party map employed.

Yu et al. (2017) propose a Street View-based urban localization method. This is a BoW-based approach in which the dictionaries are constructed using Scale Invariant Feature Transform (SIFT) and Maximally Stable Extremal Regions (MSER) features. Another Street View-based approach is the global localization method presented in Yan et al. (2018). This method relies on Oriented FAST and Rotated BRIEF (ORB)-SLAM to estimate the 3D positions of the map points. The indoor positioning by Zhou et al. (2021) proposes using ORB and Locally

Fig. 3. Perspective images downloaded from a GSV image.

Fig. 4. Pin hole model of the camera. The coordinate system of the formed image has the *x*-axis (in red) pointing right and the *y*-axis pointing down with the origin located at the top-left corner of the image.



Sensitive Hashing (LSH). Another application of GSV in localization is the metric localization of a ground robot proposed in Agarwal et al. (2015).

The navigation system proposed in Shan et al. (2015) uses Google Maps rather than Street View. This system uses Histogram of Oriented Gradients (HOG) features. However, global feature descriptors such as HOG are less robust to viewpoint changes, clutter, and occlusion.

Handcrafted feature descriptors such as SIFT, Binary Robust Invariant Scalable Keypoints (BRISK), Binary Robust Independent Elementary Features (BRIEF), and ORB are invariant to scale, rotation, and a certain degree of an affine transformation. However, they are vulnerable to significant viewpoint changes (above 45°). Due to the significant distortions in GSV images, these cannot be used directly to create visual vocabularies. In addition to that, it is known that handcrafted feature descriptors perform poorly in indoor environments, especially if the surrounding lacks prominent features and landmarks. Work in Majdik et al. (2013) has addressed the influence of distortions by generating virtual views out of the Street View images. Another author's observation was the issue of using Random Sample And Consensus (RANSAC) for outlier rejection. The authors claim that algorithms such as RANSAC work robustly, only for outlier ratios less than 50%. However, in the case of GSV, the outlier ratio can be as high as 90%. They concluded that a solution to this is to use modified versions of RANSAC such as Optimized Random Sampling Algorithm (ORSA) or Virtual Line Descriptors (VLD).

Upon the rapid developments in deep learning during the past decade, several studies have highlighted the benefits of using CNN as dense feature detectors. Maffra et al. (2018) proposes a place recognition system using a combination of 2D and 3D information. However, this method assumes that a vision-based SLAM/odometry system using a keyframe paradigm runs on a separate thread. The Multi-Domain Feature Learning (MDFL) method by Yin et al. (2019) focuses mainly on improving the robustness towards changes in environmental factors such as weather and season. However, these factors do not impact place recognition in indoor environments.

The place recognition system proposed in Maffra et al. (2019) assumes the sparse 3D map of the location corresponding to each image captured is provided by an onboard visual SLAM system. BDLoc (Li et al. 2021) proposes a global localization method that depends only on a rough latitude and longitude from a Global Positioning System (GPS). Thus, this method, unfortunately, cannot be applied to indoor scenarios.

Yu et al. (2019) proposes a loop closure detection method called Dense-Loop. This approach uses DenseNet to extract dense global features later decoupled by feature maps (decoupling by feature maps (DBF)). The authors propose using an improved version of VLAD called Weighted Vector of Locally Aggregated Descriptor (WVLAD) to encode the descriptors. WVLAD together with DBF reinforces the resistance towards scale and viewpoint changes. The study experimentally validates this through a comparison with combinations of SIFT and ORB with BoW and VLAD. Zhu and Huang (2021) propose a loop closure detection method using ShuffleNetV2. The key feature of this work is the exploitation of the comparatively low computational complexity of ShuffleNetV2 to improve execution speed. Dai et al. (2021) propose a loop closure detection method that uses pretrained VGG-16 and Resnet34 models as dense feature extractors. This method uses Kernel Principal Component Analysis (KPCA) instead of the widely used Principal Component Analysis (PCA) for dimensional reduction. However, these methods have only been evaluated on outdoor datasets. These datasets' query and reference images are also captured from the same camera.

A notable work among the loop closure detection methods evaluated indoors is LoopNet by Osman et al. (2022). The authors propose using an attention-based Siamese CNN based on ConvNet. The attention blocks incorporated into the system make the network focus more on salient regions determining key landmarks beneficial for accurate loop closure detection. The network is trained only using outdoor data. However, it performs equally in both outdoor and indoor sce-

Fig. 5. Schematic of the netVLAD CNN-based place recognition system.

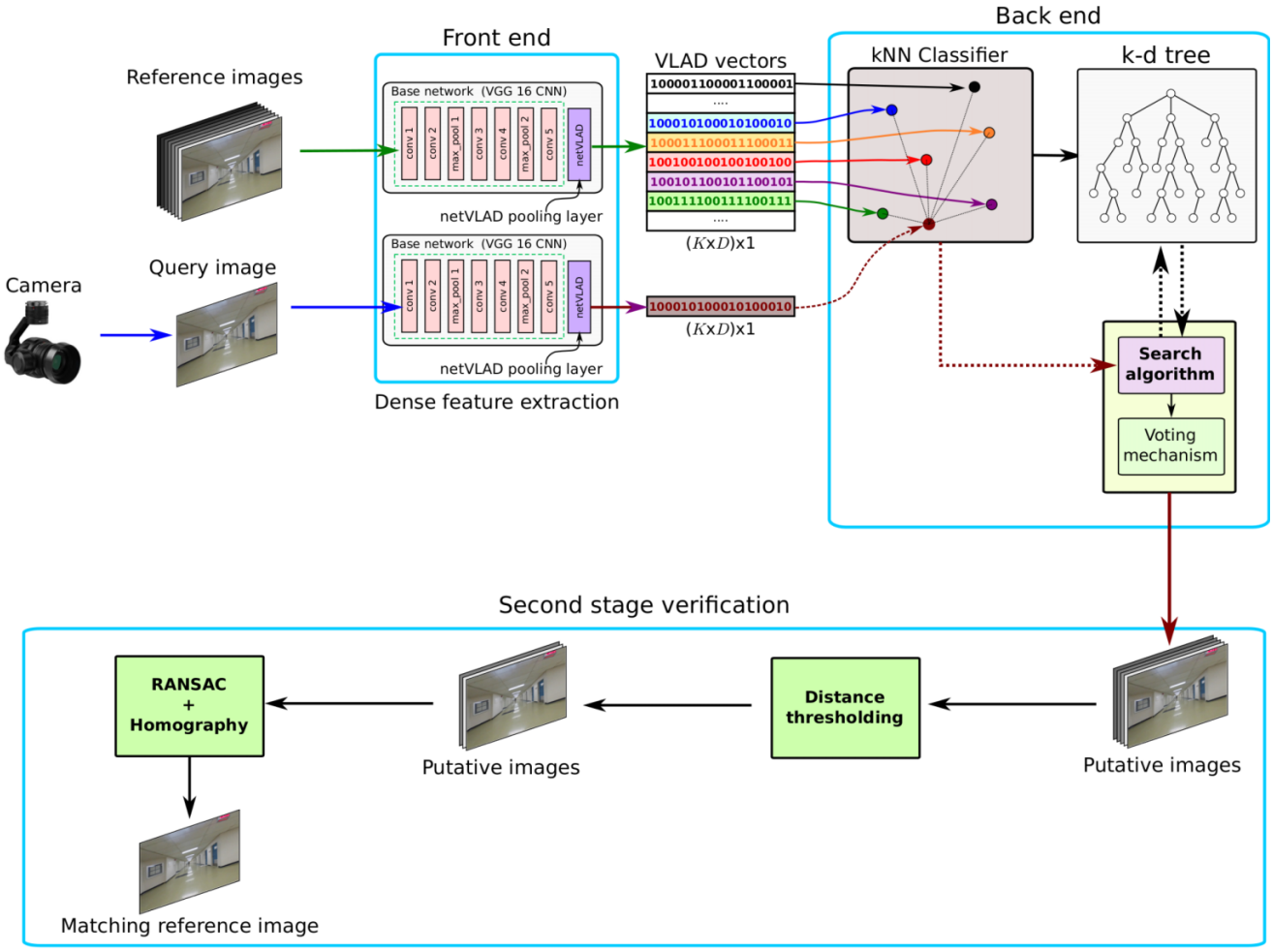


Table 2. Thresholds used for RANSAC.

Parameter	Value
Confidence threshold (%)	99.9
Maximum reprojection error (pixels)	0.0001
Maximum iterations	2000

narios. However, the datasets used for testing consist of query and reference images captured from the same camera.

A study that uses query and reference images captured from two different cameras is the indoor localization system InLoc proposed in Taira et al. (2021). In this work, the reference images are synthesized using RGBD panoramas captured through a 3D scanner, whereas the query images are from an iPhone camera. This method shows promising results in terms of localization accuracy. However, the gain in performance comes at the cost of running time, deemed a nonnegligible bottleneck by the authors.

Several attempts have been made to utilize GSV for localization. However, these are mainly outdoor studies (Agarwal et al. 2015; Yu et al. 2017). The method proposed in this paper uses visual place recognition and factor graphs together with GISV for the indoor localization of a mobile robot.

3. Methodology

3.1. Problem statement

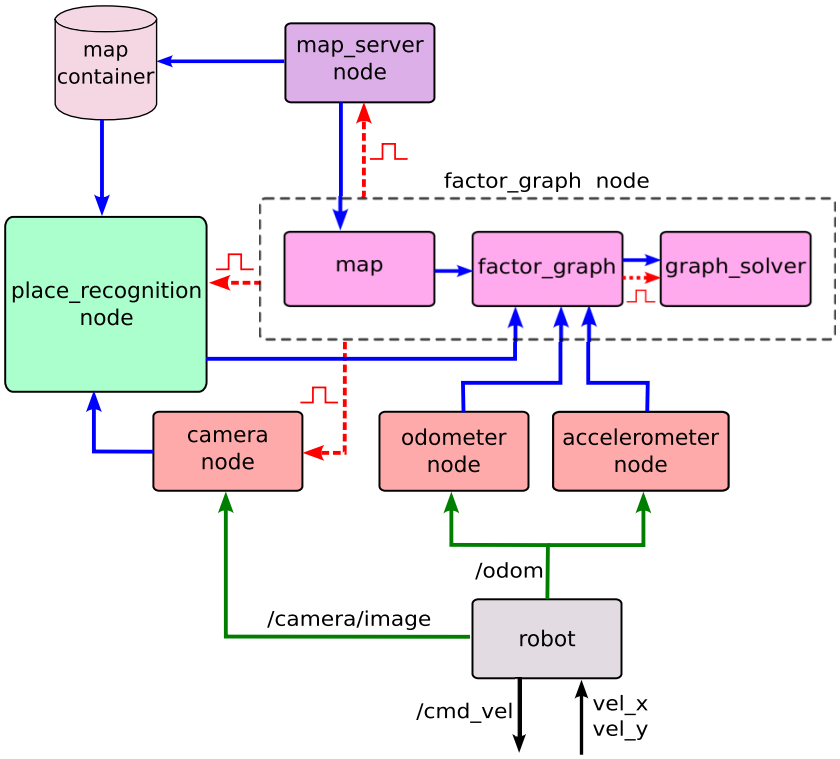
Over the years, visual place recognition has become famous for minimizing the drift in odometers in navigation systems. Recently, with studies (Tennakoon et al. 2021) concluding that CNN-based methods outperform conventional feature-based methods, CNN-based approaches have become the popular choice among many researchers as the front end of localization systems (Arandjelovic et al. 2016). Regarding navigation systems' back ends, optimization-based methods have been replacing filter-based approaches due to the many advantages of optimization methods over filter-based methods (Chen et al. 2018). With freely available maps such as GSV becoming globally available, researchers have identified the benefits of preexisting maps and have conducted studies featuring GSV-based localization (Yu et al. 2017; Yan et al. 2018). However, only a few studies have focused on using CNN-based visual place recognition for indoor environments, such as Taira et al. (2021) and Osman et al. (2022). Nevertheless, none have attempted using GISV as a preexisting map. A significant reason for this is GISV not being as widely available as GSV. Another primary reason is the lack of indoor datasets. The reference and query images are from two

Table 3. Noise values used in the factor graph.

Measurement (factor)	Noise value					
	Translation			Orientation		
	x/m	y/m	z/m	θ_x/deg	θ_y/deg	θ_z/deg
Robot's initial pose (prior factor)	0.001	0.001	0.001	0.006	0.006	0.06
Robot's camera pose (between factor)	0.001	0.001	0.1	0.001	0.001	0.001
Map camera pose (between factor)	0.001	0.001	0.1	0.001	0.001	0.001
Odometry (between factor)	0.05	0.05	0.05	0.006	0.006	0.6
Visual feedback (between factor)	0.001	0.001	0.1	0.03	0.03	0.03

Note: x, y, and z are the translations along the x, y, and z axes, respectively. θ_x, θ_y, and θ_z are rotations about the x, y, and z axes, respectively.

Fig. 6. The schematic of the factor graph-based localization system.



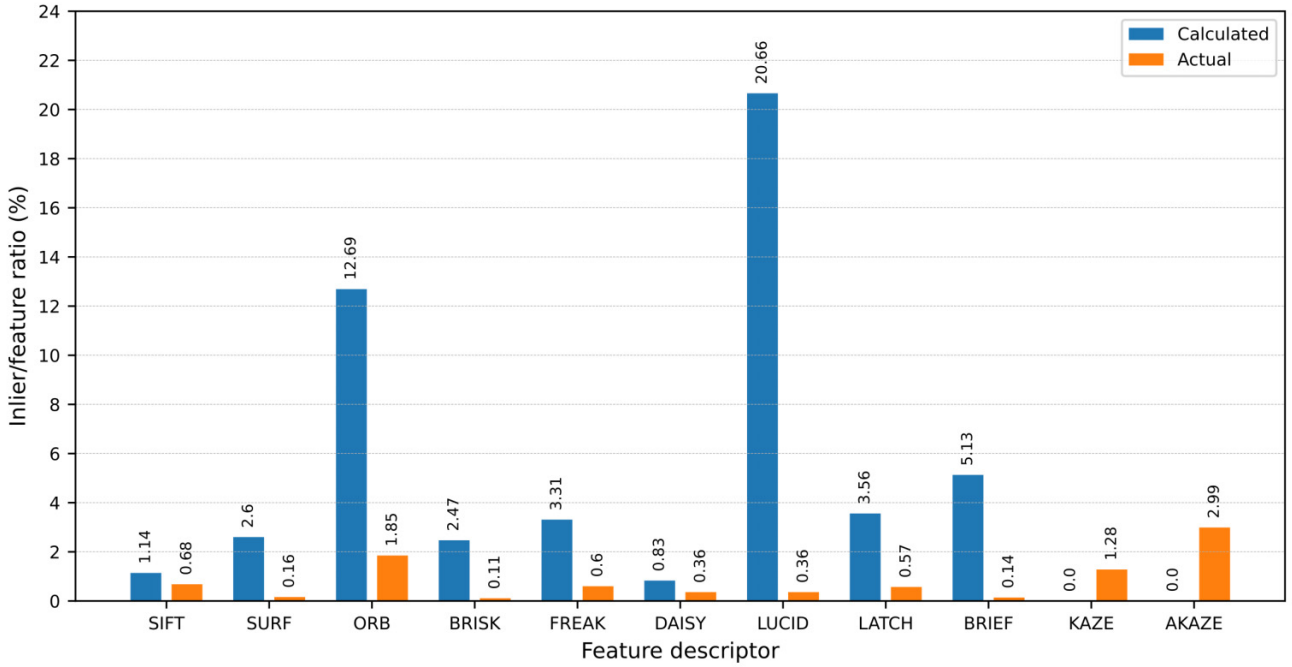
different cameras, and the unknown intrinsic matrix for reference images (GISV) makes relative pose estimation between query images and matching reference images challenging. According to our knowledge, the indoor localization system, InLoc, proposed by Taira et al. (2021) is the only study where the reference and query images captured by different cameras were used for indoor localization. However, this approach is not based on GISV.

In this work, we attempt to bridge the research gap mentioned above by designing a GISV-based place recognition system to localize a mobile robot in an indoor environment. We propose using GISV for several reasons. First, it is freely available. Second, the proposed place recognition system can be easily reconfigured to use any map available on GISV. Third, Google allows users to create custom indoor Street View maps and upload them to their servers.

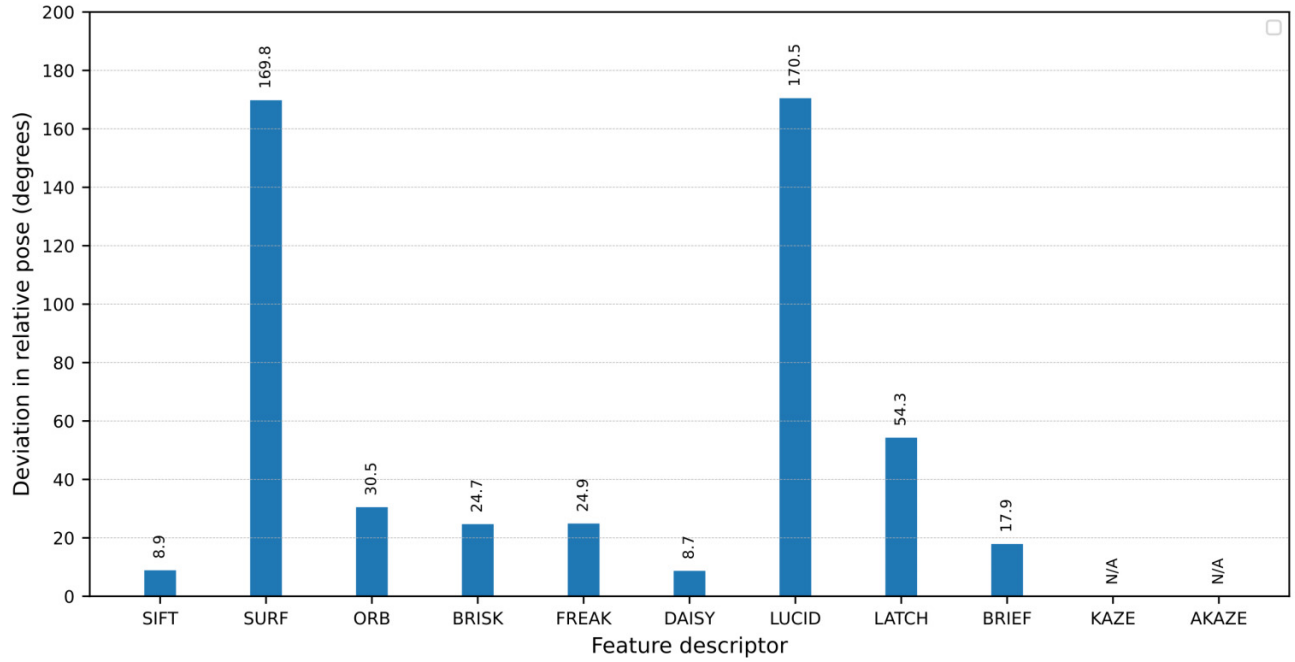
The key features of the design are as follows.

- The place recognition system should be able to handle indoor images with a lack of visually distinct features and viewpoint and illumination changes between images taken at the same location.
- The place recognition system should utilize GISV images as its reference images and use the images from the robot's onboard camera as query images, thereby being able to match images from two dissimilar cameras.
- The place recognition system should have a suitable virtual view generation method to support matching, and an equivalent camera matrix generation method should be present to support localization algorithms.
- The resulting image matches from the place recognition module should be used to correct the drift of the robot in a consistent way to support the robot's navigation.
- The developed system should be experimentally verified and optimized for online use with mobile robots.

Fig. 7. Calculated inlier ratio, actual inlier ratio, and deviation in up-to-scale relative pose versus the type of features used.



(a) Calculated and actual inlier/feature ratio vs the type of features



(b) Deviation in relative pose vs the type of features

This section details the key components of the proposed localization system, starting with the custom reference dataset gathered for the evaluation of the proposed system.

3.2. Map of MUN Engineering building basement

The site chosen to conduct the experiments was the Memorial University of Newfoundland (MUN) Engineering building basement. Thus, a GISV map of the basement was necessary to construct the reference images required for place recognition. As yet, GSV for building interiors is not readily available as it is for outdoors. However, Google has provided a platform through which consumers can construct custom indoor Street View maps and upload them to their servers. Using this platform, a GISV map of the MUN Engineering basement corridor was created. It comprises forty-nine 360° images (Fig. 1) that were captured using a Samsung Gear 360 camera. These images were captured at preselected locations along the MUN Engineering basement corridor, which are approximately

Table 4. Values and selection criteria of design parameters.

Module	Parameter	Value	Selection criteria
Place recognition	Reference images	–	Hand-picked, considering how unique the corresponding physical location is. Locations such as junctions and lounge areas were picked as suitable locations. The automatic selection of these images is not addressed in the scope of this work
	Image size	640 × px 640 px	The maximum resolution of the images that could be downloaded from the GISV API
	FOV of the images	90°	Captures a higher visual field without introducing distortions in the downloaded images
	Number of bins in netVLAD	64	Used the default value used by the original authors of Arandjelovic et al. (2016). The value is high enough to maintain the accuracy of predictions without increasing computational cost and prediction time
	Number of candidate matches	5	Empirically determined to improve the chances of the correct match becoming a candidate for the second-stage verification without compromising the prediction time
	Distance threshold	1.275	Empirically determined to reduce the false positive rate to zero. It requires fine-tuning when new datasets are introduced to the system
	RANSAC algorithm	5-point	The 5-point algorithm is faster and more robust than the 8-point algorithm (Nistér 2004)
	Confidence threshold for RANSAC	0.99	This is the desired probability of successfully detecting inliers. 99% is the most commonly used value in literature and widely used computer vision libraries
	Maximum reprojection error	0.0001	Empirically determined to reduce the number of inliers filtered out while rejecting outliers
	Minimum number of inliers	10	It was noticed during experiments that correct matching reference and query images contained more than 10 inliers, whereas incorrect matches had fewer than 10
	Similarity threshold	0.02	Empirically determined to filter out any remaining false positive predictions
Factor graph	Minimum translation of the robot required to add a node to the factor graph	5 m	Empirically determined as a trade-off between the complexity of the factor graph and the accuracy of the solved path. Lowering this value increases the number of nodes in the factor graph, increasing the complexity and computational overhead in solving the graph but improving the accuracy of the solved path and vice versa
	Minimum rotation of the robot required to add a node to the factor graph	45°	The path the robot traversed during the experiments only had 90° bends. When 45° is used as the minimum rotation required, a node is added to the factor graph at every corner. This improves the factor graph's resemblance to the robot's actual trajectory
	Noise values of the factors	Refer Table 3	Determined empirically. *Z*-axis in the robot's coordinate frames and map cameras corresponded to depth. Hence, the translation noise in the *z*-direction of the visual feedback measurements was tuned to a higher value to overcome the relative translation being up to a scale

4.5 m (15 feet) from one another. The preselected locations later become the nodes for the topological map used by the factor graph implementation. Since the nodes were selected premeditatively, the approximate positions of the nodes relative to the map's origin are known. The map's first node (node 1) was the origin. For simplicity, the absolute position of the origin was set to be (0,0,0). Shown in Fig. 2 is an illustration of the aforementioned topological map.

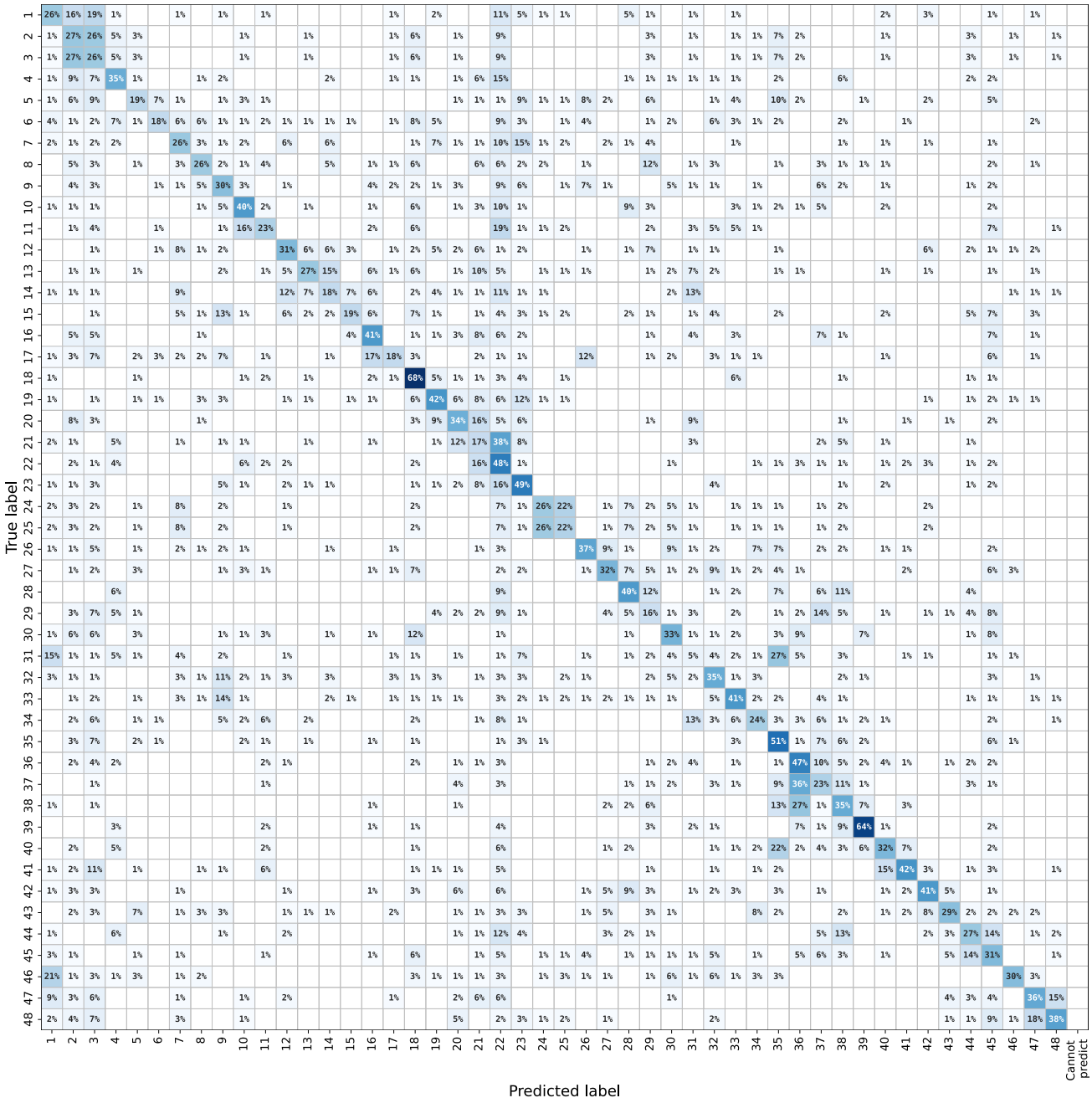
3.3. Creating the reference image database

Images necessary to construct the reference dataset of the MUN Engineering basement corridor were extracted using the GSV API. The API allows the users to download perspective images 640 px × 640 px in size from a GSV image at the desired heading, pitch, and Field Of View (FOV). Six perspective images were downloaded from each Street View image using the values given in Table 1. Figure 3 shows a few examples of the perspective images downloaded.

3.4. Determining equivalent intrinsic matrix

The intrinsic parameters of the cameras from which they were captured are needed to estimate the relative pose between a query image and a reference image. Since the query images come from a camera mounted on the robot, its intrinsic parameters are known using a camera calibration procedure (Zhang 2000). However, the reference images are perspective images generated from the Street View images. Hence, the intrinsic parameters of the 360° camera used to capture the street view images no longer apply to them. Therefore, equivalent intrinsic parameters are derived for the reference images.

Fig. 8. Confusion matrices (48 × 49) at each stage of place recognition.



(a) Confusion matrix before distance thresholding

Assume that the equivalent camera can be modelled as a pinhole camera as shown in Fig. 4. Here, θ_W and θ_H are the horizontal and vertical FOV, respectively, while f is the focal length. W and H are the width and height of the image formed, respectively. (c_x, c_y) is the coordinates of the image centre with reference to the image coordinate system shown (Corke 2017).

Then, the equivalent intrinsic matrix, K_{eq}, can be represented as shown by eq. 1:

$$K_{eq} = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}_{3\times3} \tag{1}$$

where

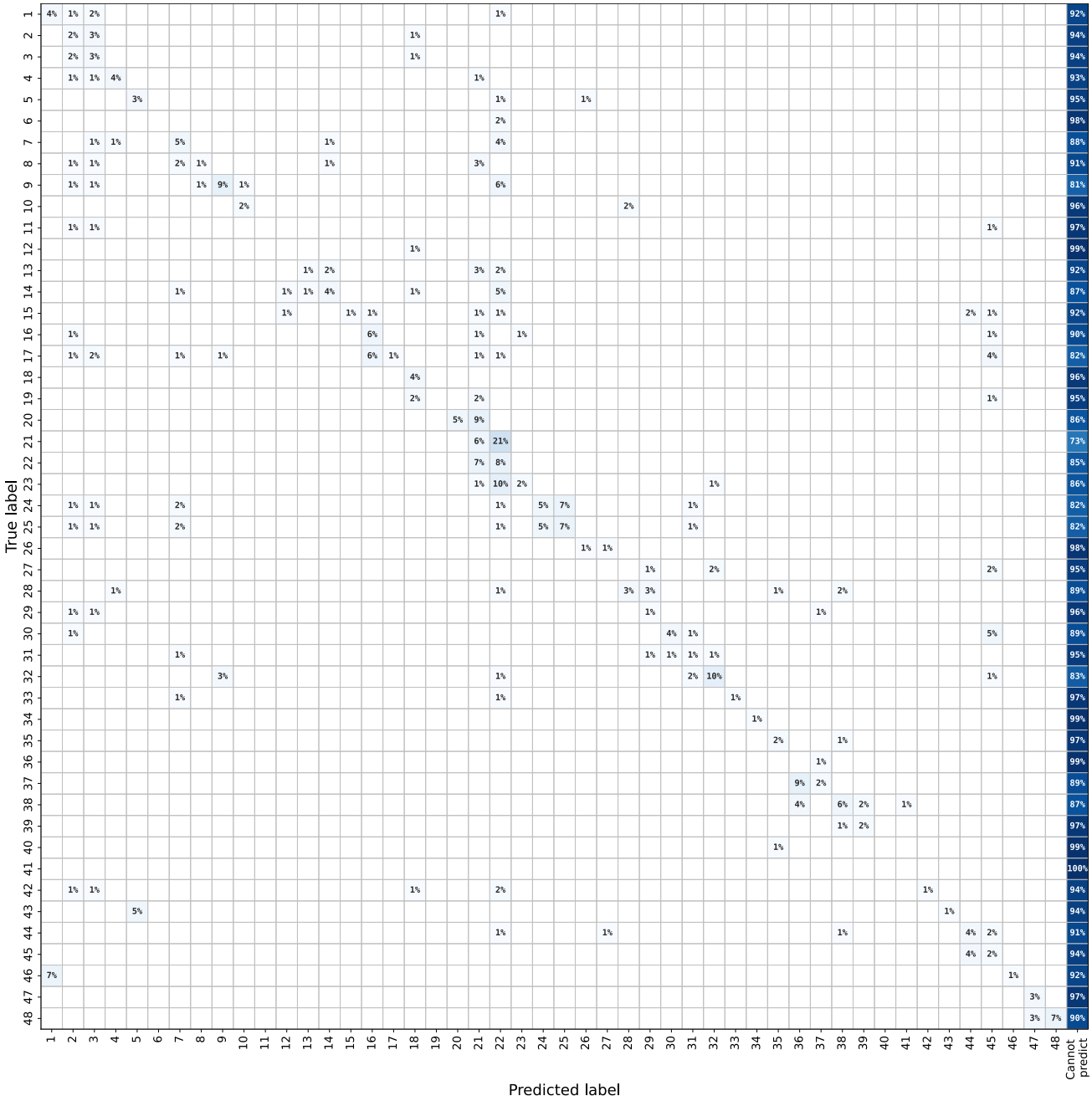
$$c_x = \frac{W}{2} \tag{2}$$

$$c_y = \frac{H}{2} \tag{3}$$

$$f = \frac{H}{2 \cdot \tan\left(\frac{\theta_H}{2}\right)} = \frac{W}{2 \cdot \tan\left(\frac{\theta_W}{2}\right)} \tag{4}$$

θ_H and θ_W are parameters available when downloading the images using GSV API. For maximum FOV without distor-

Fig. 8. (*Continued.*)



(b) Confusion matrix after distance thresholding and before geometric similarity check

tions

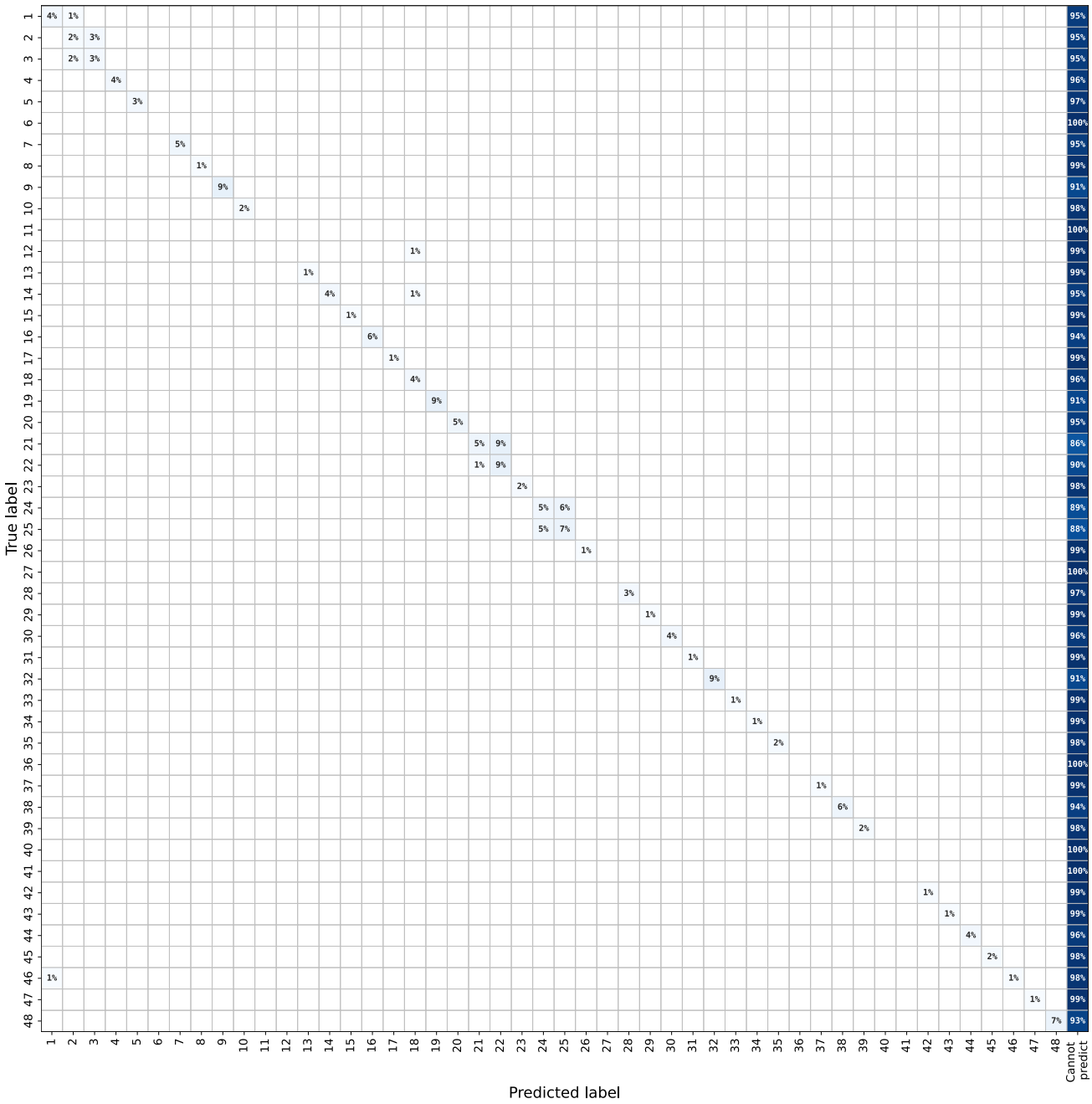
$$\frac{\pi}{3} \leq \theta_{\mathrm{H}}, \theta_{\mathrm{W}} \leq \frac{2\pi}{3} \tag{5}$$

3.5. CNN-based place recognition system

The CNN-based place recognition system comprises three main parts, i.e., the front end, back end, and second-stage verification (Fig. 5). The front end, which serves as the dense feature extractor, comprises a base network with the first seven layers of a VGG-16 off-the-shelf CNN pretrained on an ImageNet dataset, appended at the end with a netVLAD pooling layer. The base network extracts the normalized dense features of the images. The netVLAD layer clusters the features into 64 bins and computes the centroid of each bin. These centroids serve as the visual words necessary to derive the VLAD vector of each image. Then, a VLAD vector is generated for each image using the computed visual words and dense features. Each VLAD vector encodes a unique representation of its corresponding image.

The back end comprises the nearest neighbour (NN) classifier, a data structure and a search and vote mechanism. The NN classifier was trained offline using the VLAD vectors and the corresponding labels of the reference dataset. During the online phase, the VLAD vector of the query image is passed to an NN classifier, which predicts a set of candidate images that best matches the query image.

Fig. 8. (*Concluded.*)



(c) Confusion matrix after distance thresholding and geometric similarity check

The system is also composed of a two-step second-stage verification module. The first step calculates the distances between the query and candidate reference images. Candidates with a distance greater than a threshold are filtered out. The second step is a geometric similarity check. A RANSAC-based homography check is conducted on each candidate image that passes the first step and determines the number of inliers. If the number of inliers exceeds a minimum threshold, the corresponding candidate image is considered a good match for the query image. Ten was empirically determined as a suitable minimum threshold for the number of inliers. The parameters and values used for RANSAC are presented in Table 2.

For those candidate images that qualify, a similarity score is calculated using eq. 6:

$$S_{\text{similarity}} = \frac{2 \times N_{\text{inliers}}}{N_{\text{matches}} + N_{\text{images}}} \tag{6}$$

where $S_{\text{similarity}}$ is the similarity score, N_{inliers} is the number of inliers between a candidate image and the query image, N_{matches} is the number of feature matches between the candidate image and the query image, and N_{images} is the number of candidate images.

The candidate images that score less than a predetermined threshold are filtered out. Out of the remaining, the node corresponding to the candidate reference image, which records the highest similarity score, is selected as the node closest to the considered query image.

Key steps of the place recognition module are shown in Algorithm 1.

Algorithm 1 Place Recognition

Require: $I_{ref} = \{r_1, r_2, \ldots, r_n\}$ the set of reference images
$weights = \{w_1, w_2, \ldots, w_m\}$ the weights for the netVLAD CNN
$V_{ref} = \{v_1, v_2, \ldots, v_n\}$ the set of VLAD vectors of reference images
$L_{ref} = \{l_1, l_2, \ldots, l_n\}$ the set of image labels of reference images
$I_{query} = \{q_1, q_2, \ldots, q_i, \ldots, q_n\}$ the query images

(—- Offline phase —-)

netVLAD CNN $\xleftarrow{compile\ model}$ $weights$
NN classifier $\xleftarrow{train}$ V_{ref}, L_{ref}

(—- Online phase —-)

for $q_i \in I_{query}$ **do**
  $vlad \xleftarrow{generate\ VLAD}$ **netVLAD CNN** $\leftarrow q_i$
  $PRED$, $DIST \xleftarrow{predict}$ **NN classifier** $\leftarrow vlad$ ($PRED \subset I_{ref}$)
  for all $p \in$ **PRED do**
    if $DIST[p] \geq dist\_thresh$ **then**
      Reject p
    end if
  end for
  for all $p \in$ **PRED do**
    $d_p \leftarrow$ SIFT of p
    $d_q \leftarrow$ SIFT of q_i
    matches$\{p\} \leftarrow$ Ratio test $\leftarrow$ Match d_p and d_q
  end for
  for all $p \in$ **PRED do**
    inliers $\leftarrow$ **RANSAC + Homography** $\leftarrow$ matches
    if inliers ≥ 10 **then**
      Compute $S_{similarity}$
      if $S_{similarity} \geq similarity\_thresh$ **then**
        best_match $\leftarrow p$
      else
        Reject p
      end if
    end if
  end for
  return $p \in$ best_match with $max(S_{similarity})$
end for

The process begins with an offline phase. During this phase, the weights of the netVLAD CNN model are loaded, and the model is compiled. In addition, the NN classifier is trained using the preexisting VLAD vectors of the reference images and the corresponding labels. During the online phase, the netVLAD CNN generates the VLAD vector for each query image. The NN classifier takes this VLAD vector as the input and predicts the five best matching reference images along with a distance metric of each match. Next, the predicted matches whose distance metric is lower than a predetermined threshold are rejected. This threshold is tuned using the dataset. Then, for each of the remaining predicted matches, the SIFT feature descriptors are extracted and matched with the SIFT feature descriptors of the query image. Using the matched SIFT descriptors, the geometric consistency between the query image and each predicted match is checked using homography and RANSAC-based outlier rejection. After that, a similarity score is calculated for the predicted matches that result in a minimum of 10 inliers. The predicted matches with a similarity score below a predetermined threshold are rejected. Finally, out of the remaining predicted matches, the match that records the highest similarity score is determined as the reference image that best matches the query image.

3.6. Factor graph module

The approach assumes a preconstructed map, prior knowledge of the robot's initial position, and the existence of odometry measurements, either by wheel encoders or a Visual-Inertial Navigation System (VINS). The robot's initial position and the positions of the map nodes contribute to the factor graph as prior factors. The odometry measurements, on the other hand, contribute as between factors. The noise model of each factor (measurement) was chosen to be Gaussian. The noise values for each factor were determined empirically. The noise values used for the experiments are listed in Table 3.

Algorithm 2 illustrates the critical processes of the factor graph module.

Algorithm 2 Factor Graph

Require: **Map** - the set of map poses
Odom - Odometry data
Cam - Camera image feed
Noise_Models - Noise models of **Map**, **Odom** and **Cam**

create non-linear ***factor_graph***
factor_graph $\xleftarrow{prior\ factor}$ **Map** nodes
factor_graph $\xleftarrow{prior\ factor}$ Robot initial pose
initialize ***factor_graph***
$N_{robot\ poses} \leftarrow 0$
while (**Odom** & **Cam** & **Noise_Models** are available) **do**
  d $\leftarrow$ **Odom**
  $\theta \leftarrow$ **Odom**
  if (d $\geq d_{thresh}$ **or** $\theta \geq \theta_{thresh}$) **then**
    factor_graph $\xleftarrow{Pose}$ Robot pose
    factor_graph $\xleftarrow{between\ factor}$ (d,θ)
    initialize ***factor_graph***
    $N_{robot\ poses} \leftarrow N_{robot\ poses} + 1$
  end if
  if ($N_{robot\ poses} = 5$) **then**
    image $\leftarrow$ **Cam**
    Place Recognition System $\leftarrow$ image
    Essential_Mat $\leftarrow$ ***Place Recognition System***
    ${}^{Rob\_cam}\boldsymbol{Pose}_{Map\_cam} \Leftarrow$ Essential_Mat
    factor_graph $\xleftarrow{Betweenfactor}$ ${}^{Rob\_cam}\boldsymbol{Pose}_{Map\_cam}$
    calculated robot poses $\leftarrow$ solve ***factor_graph***
    return calculated robot poses
    $N_{robot\ poses} \leftarrow 0$
  end if
end while

While the robot is in motion and sensor information is available, nodes are frequently added to the factor graph based on two criteria. Either the robot has to travel a thresh-

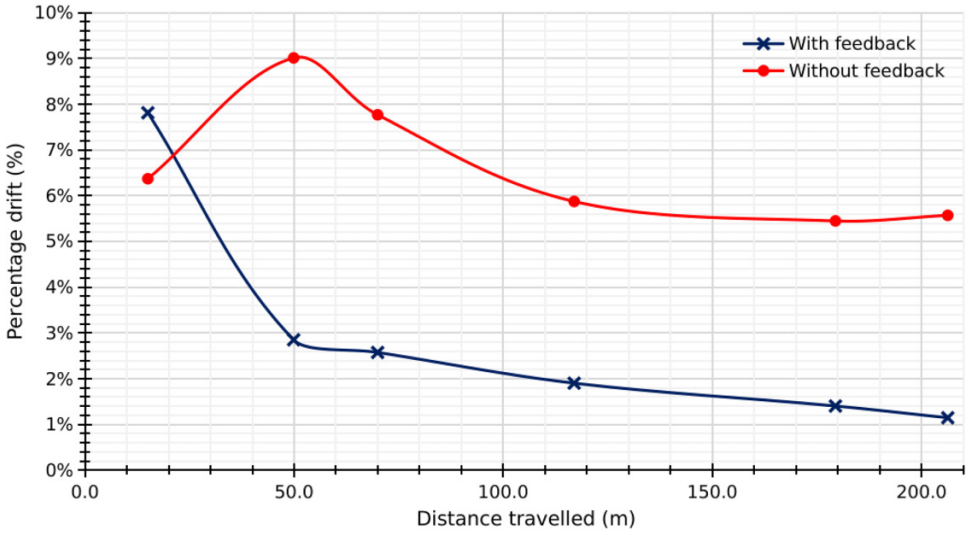
Fig. 9. Prediction results before distance thresholding.

Fig. 10. Prediction results after distance thresholding and before second-stage verification.

old distance or turn by a threshold angle relative to its preceding node. Consecutive nodes on the factor graph are connected through odometry measurements introduced into the factor graph as between factors. At predefined intervals, an image is captured through the robot's camera and input to the place recognition system (Section 3.5). If the place recognition system predicts the map node closest to the robot's current position, the relative pose between the robot's camera and the map camera is calculated. It is accomplished by decomposing the essential matrix between the query image

Fig. 11. Prediction results after distance thresholding and second-stage verification.

Fig. 12. Percentage drift versus distance travelled.



and the best-matching reference image determined through homography. This relative pose is introduced into the factor graph as a factor connecting the robot's pose and the corresponding map node. The relative translation being up to a scale was overcome by tuning the noise value of visual feedback measurements corresponding to the translation. *Z*-axis was the coordinate axis corresponding to depth in the coordinate frames of both the robot and map cameras. Thus,

Table 5. Position and orientation RMSE with and without visual feedback.

	RMSE	
Test case	Position/m	Orientation/deg
Without visual feedback	7.02	3.35
With visual feedback	1.56	2.05

the translation noise in the *z*-direction of the visual feedback measurements was tuned to a higher value. After each successful integration of a visual feedback constraint, the factor graph is solved, resulting in the corrected trajectory of the robot. The cameras' intrinsic matrices are needed to decompose the essential matrix into relative poses. The robot's camera's intrinsic matrix is known from calibration. As the intrinsic matrix of the map camera, the equivalent intrinsic matrix derived in Section 3.4 is used.

The system was implemented in Robot Operating System (ROS) with the help of Georgia Tech Smoothing and Mapping (GTSAM) and OpenCV libraries. The critical processes in the factor graph process were implemented as ROS nodes. It enabled those processes to operate in parallel, independent of one another. The ROS nodes were connected through ROS topics. Important repetitive tasks, such as adding factors and solving the factor graph, were implemented as ROS services, which, when necessary, can be triggered via service requests. Figure 6 illustrates the architecture of the factor graph module as implemented in ROS. The system's modular nature proved helpful for the development and debugging of the system.

3.7. Feature detection and matching

Work in Tennakoon et al. (2021) has compared several conventional feature descriptor types for an indoor dataset to determine the best feature type for visual place recognition. The authors have shown that considering both the actual inlier ratio and the deviation in the up-to-scale relative pose with a benchmark as evaluation metrics, the SIFT descriptor has produced the lowest deviation at the highest actual inlier ratio. Their main results are summarized in Fig. 7. The results suggest that among the feature types considered, SIFT reports the best performance. Hence, in this study, SIFT features were used for feature detection and matching in the place recognition module.

3.8. Design parameter selection

The overall system results in several design parameters. The values of the important design parameters and the selection criteria are listed in Table 4.

3.9. Experimental setup

A ROS bag was collected by navigating a Seekur Jr. robot through the MUN Engineering basement corridor. This ROS bag was used to conduct experiments offline. The distance and angle thresholds for the factor graph, i.e., the minimum distance and angle by which the robot should move relative to the preceding node for a new node to be added to the factor graph, were arbitrarily fixed at 5 m and 45°, respectively. The ground truth of the robot was estimated using the laser scan data from the ROS bag and a particle filter from the ROS-amcl package. The place recognition module and the factor graph module were tested separately for evaluation purposes. The relative poses necessary to implement the visual feedback constraints were computed separately and integrated into the factor graph.

4. Results

4.1. CNN-based place recognition system

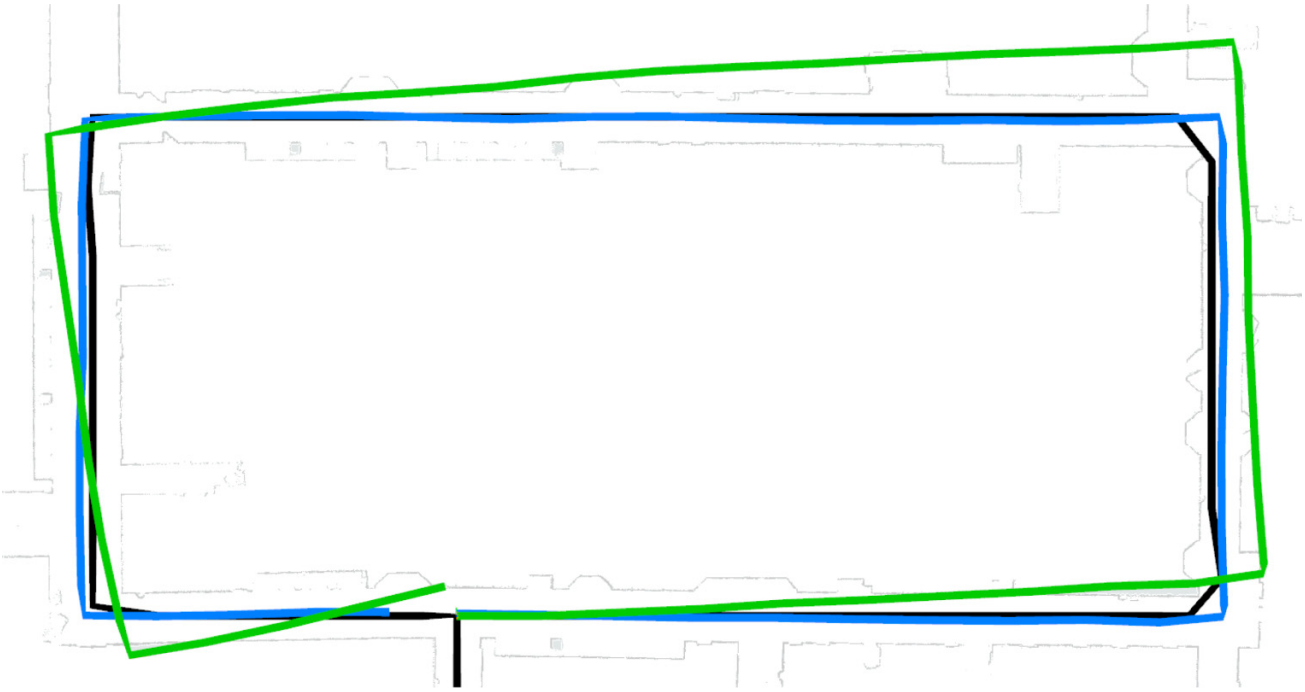
Figure 8 is the confusion matrices of the place recognition system for the MUN Engineering basement dataset. The confusion matrices illustrate how the distance thresholding and geometric similarity check reduce the number of false positive predictions to zero. False positive predictions introduce incorrect visual feedback constraints into the factor graph, resulting in incorrect solutions to the factor graph. A single false positive prediction can compromise the accuracy of the solution to the entire factor graph. On the other hand, a single correct prediction can significantly improve the overall accuracy of the solution to the factor graph. Thus, it is crucial to maintain a zero false positive prediction rate.

Figure 8*a* shows the confusion matrix for the predictions before distance thresholding or geometric similarity check. It can be seen that a considerable percentage of predictions lie on the leading diagonal of the confusion matrix. However, a substantial percentage of predictions also appear in the off-diagonal positions. It can be seen that the total percentage of off-diagonal predictions along each column of the confusion matrix is higher than the percentage of on-diagonal predictions. It indicates that the place recognition system without second-stage verification has a higher false positive rate. Prediction results before distance thresholding are shown in Fig. 9.

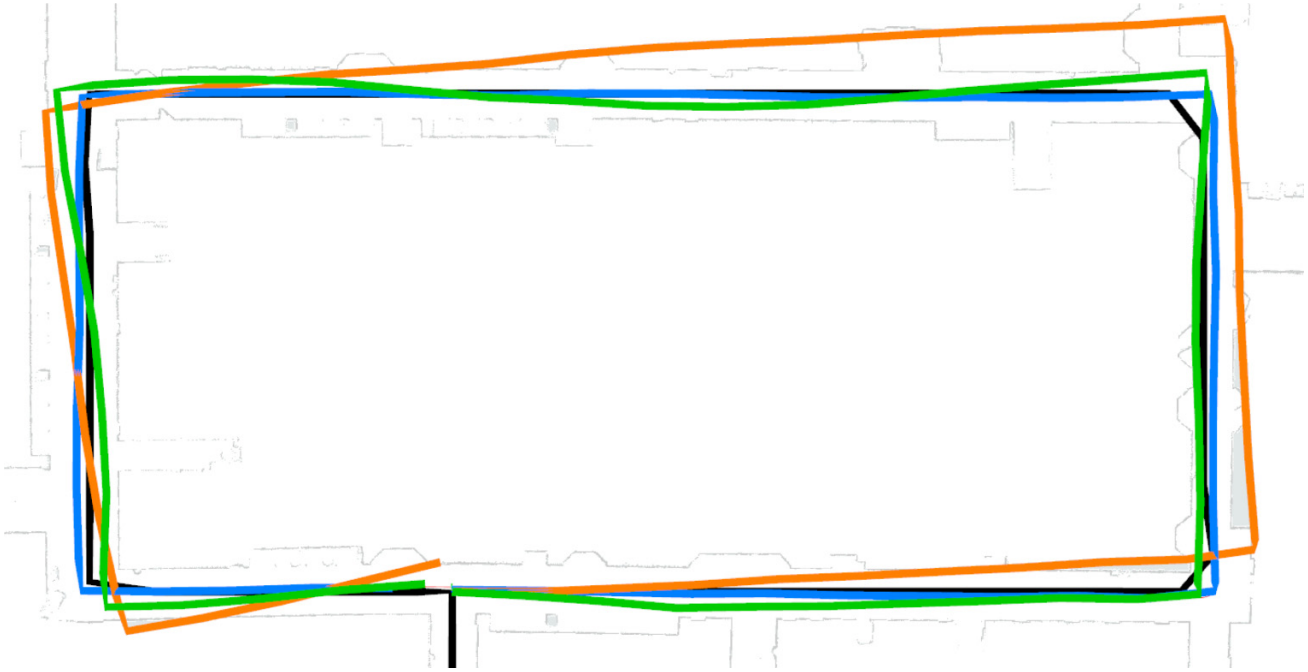
Figure 8*b* shows the confusion matrix after distance thresholding but before the geometric consistency check. It can be seen that a majority of the off-diagonal predictions have been categorized as "cannot predict" and moved to the last column of the confusion matrix. "Cannot predict" means that the place recognition system cannot make a significantly accurate prediction for a considered query image. Hence, "cannot predict" does not qualify as an incorrect prediction either. As a result, the false positive rate of predictions gets significantly reduced. However, distance thresholding alone has not eliminated all the off-diagonal predictions. It means that although the false positive rate is reduced, it still is not zero. Prediction results after distance thresholding but before the geometric consistency check are shown in Fig. 10.

Figure 8*c* shows the confusion matrix after both distance thresholding and geometric similarity check. It can be seen that almost all the off-diagonal predictions have been categorized as "cannot predict". It ensures that the false positive rate of predictions has been reduced to almost zero. However, few off-diagonal predictions remain without being filtered out. Some of the off-diagonal predictions are adjacent to on-diagonal predictions. These correspond to locations that

Fig. 13. Solution to the factor graph with and without visual feedback.



(a) Solution to the factor graph without visual feedback. Ground truth(blue), map(black) and the solution(green)



(b) Solution to the factor graph with visual feedback. Odometry(orange), ground truth(blue), map(black) and the solution(green)

are physically only a couple of metres apart and look visually similar. Other off-diagonal predictions correspond to physical locations that are a few more metres apart and look visually similar (true labels 1 and 46 correspond to the start of the loop and a few metres before the end of the loop, which is the same as the starting point). Both are cases of perceptual aliasing, physically distant locations appearing visually similar. It is an inherent challenge experienced in place recog-

nition. However, this can be solved to a certain extent when selecting reference images by eliminating visually similar images corresponding to physically distant locations. Prediction results after both distance thresholding and geometric similarity are shown in Fig. 11.

4.2. Factor graph module

Figure 12 shows the drift in position as a percentage of the total distance travelled against the distance travelled by the robot. In the absence of visual feedback, the percentage drift starts high, increases, and then drops to a steady value of about 5.5%. On the other hand, in the presence of visual feedback, the percentage drift starts high but decreases at a steady rate with each visual feedback constraint introduced into the factor graph. Even towards the end of the loop, the percentage drift in the presence of visual feedback shows a noticeable downwards trend.

As seen in Table 5, while having visual feedback, the Root Mean Square Error (RMSE) in position has experienced a significant reduction compared with not having visual feedback. The RMSE in orientation in the presence of visual feedback too is lesser than in the absence of it, although the reduction is not very pronounced as in the case with the position.

The improvement in the optimized trajectory in the presence of visual feedback constraints can be seen in Fig. 13. As seen in Fig. 13*a*, the optimized trajectory (shown in green) is well off from the ground truth (shown in blue). On the other hand, in Fig. 13*b*, the optimized trajectory with visual feedback constraints is much closer to the ground truth. It suggests that the factor graph-based localization system, together with the CNN-based place recognition, is effective in improving the localization accuracy of a mobile robot platform.

5. Conclusions

This paper proposes a mobile robot localization system developed by employing GISV and a CNN-based place recognition system. The proposed localization system comprises two main modules: a place recognition module based on a netVLAD-based CNN and a factor graph-based optimization module.

This study developed and tested the CNN-based place recognition and factor graph modules. A custom Indoor Street View map of the MUN Engineering building basement was created and uploaded to the GSV servers. The reference images for the place recognition module were downloaded from this GISV map using an API. The reference images were handpicked, considering the physical locations. Unique locations such as junctions were chosen as landmarks to be included in the reference image set, reducing the possibility of perceptual aliasing. The performance of the place recognition system is presented using confusion matrices. The results indicated that reliable indoor visual place recognition using GISV is possible. The results also show how the two-step second-stage verification module composed of distance thresholding and geometric consistency check can improve prediction accuracy by driving the false positive rate to zero.

The factor graph was experimentally validated using a custom dataset captured at the MUN Engineering building basement using a Seekur Jr. robot. The factor graph exhibited nondrifting performance. The solution to the factor graph with visual feedback constraints was in close agreement with the ground truth. The results presented were generated using only six visual feedback constraints. The visual feedback constraints reduce the localization error in both position and orientation with RMSE in position and orientation less than 2 m and 2.5°, respectively. In addition to the above, the results also showed that visual loop closure using reference and query images obtained from dissimilar sources could improve the localization accuracy of a factor graph-based localization system.

6. Future work

Incorporating the CNN-based place recognition module and the factor graph module into a single unified system is a potential improvement towards a fully automated navigation system. In addition, introducing a landmark identification module to the overall system would further improve the system's autonomy.

Furthermore, applications of the localization system in automating tasks such as indoor patrol and creating indoor street view maps are some other areas that are worth investigating. Additionally, applications involving aerial imagery are another promising area for future research.

Nevertheless, there are several possible future challenges. Automatically selecting suitable images for the reference image dataset and determining effective means of handling false positive matches entering the localization module are some challenges that need consideration.

Ultimately, developing a GISV and factor graph-based fully autonomous indoor localization system would open numerous avenues in the future, such as building-wide localization, indoor mapping, indoor patrolling, security, and surveillance. The ease of extending to use of a multitude of sensors, efficient use of hardware, reduced requirement for tuning, ability to easily migrate between indoor and outdoor environments, and ability to effectively utilize sensor information available at any given time can be seen as advantages of using such a system for indoor localization.

Article information

History dates

Received: 6 September 2022
Accepted: 24 February 2023
Accepted manuscript online: 20 March 2023
Version of record online: 11 April 2023

Data availability

Data generated or analyzed during this study are available in the Mendeley Data repository, reserved DOI:10.17632/42fsy3ncby.1, https://data.mendeley.com/v1/datasets/42fsy3ncby/draft.

Author information

Author ORCIDs

Kusal B. Tennakoon https://orcid.org/0000-0002-2193-4553

Author contributions

Conceptualization: KBT, ODS
Data curation: KBT
Formal analysis: KBT
Investigation: KBT
Methodology: KBT
Resources: ODS, AJ, GKIM, RGG
Software: KBT
Supervision: ODS, AJ, GKIM, RGG
Validation: KBT, ODS
Visualization: KBT
Writing – original draft: KBT
Writing – review & editing: KBT, ODS, AJ, GKIM

Competing interests

The authors declare that there are no competing interests.

Funding information

The authors declare no specific funding for this work.

References

Agarwal, P., Burgard, W., and Spinello, L. 2015. Metric localization using Google Street View. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3111–3118. doi:10.1109/IROS.2015.7353807.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5297–5307.

Ascani, A., Frontoni, E., Mancini, A., and Zingaretti, P. 2008. Feature group matching for appearance-based localization. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. pp. 3933–3938.

Bresson, G., Yu, L., Joly, C., and Moutarde, F. 2019. Urban localization with street views using a convolutional neural network for end-to-end camera pose regression. In IEEE Intelligent Vehicles Symposium, Proceedings, Vol. 2019. Institute of Electrical and Electronics Engineers Inc. pp. 1199–1204. doi:10.1109/IVS.2019.8813892.

Chen, C., Zhu, H., Li, M., and You, S. 2018. A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives. Robotics, **7**: 45. doi:10.3390/robotics7030045.

Corke, P. 2017. Robotics, vision and control: fundamental algorithms in MATLAB®. 2nd, completely revised, extended and updated ed. Vol. **118**. Springer.

Dai, K., Cheng, L., Yang, R., and Yan, G. 2021. Loop closure detection using KPCA and CNN for visual SLAM. In Chinese Control Conference, CCC. IEEE Computer Society, Vol. **2021**. pp. 8088–8093. doi:10.23919/CCC52363.2021.9550432.

Kejriwal, N., Kumar, S., and Shibata, T. 2016. High performance loop closure detection using bag of word pairs. Robot. Auton. Syst. **77**: 55–65.

Li, H., Fan, T., Zhai, H., Cui, Z., Bao, H., and Zhang, G. 2021. BDLoc: global localization from 2.5D building map. In Proceedings of the 2021 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2021. Institute of Electrical and Electronics Engineers Inc. pp. 80–89. doi:10.1109/ISMAR52148.2021.00022.

Maffra, F., Chen, Z., and Chli, M. 2018. Viewpoint-tolerant place recognition combining 2D and 3D information for UAV navigation. In Proceedings of the IEEE International Conference on Robotics and Automation. Institute of Electrical and Electronics Engineers Inc. pp. 2542–2549. doi:10.1109/ICRA.2018.8460786.

Maffra, F., Teixeira, L., Chen, Z., and Chli, M. 2019. Real-time wide-baseline place recognition using depth completion. IEEE Robot. Autom. Lett. **4**(2): 1525–1532. doi:10.1109/LRA.2019.2895826.

Majdik, A.L., Albers-Schoenberg, Y., and Scaramuzza, D. 2013. MAV urban localization from Google Street View data. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. pp. 3979–3986.

Morel, J.M., and Yu, G. 2009. ASIFT: a new framework for fully affine invariant image comparison. SIAM J. Imaging Sci. **2**(2):438–469. doi:10.1137/080732730.

Nistér, D. 2004. An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**: 756–770.

Osman, H., Darwish, N., and Bayoumi, A.E.M. 2022. LoopNet: where to focus? Detecting loop closures in dynamic scenes. IEEE Robot. Autom. Lett. **7**(2):2031–2038. doi:10.1109/LRA.2022.3142901.

Piasco, N., Sidibe, D., Gouet-Brunet, V., and Demonceaux, C. 2019. Learning scene geometry for visual localization in challenging conditions. In Proceedings of the IEEE International Conference on Robotics and Automation. Institute of Electrical and Electronics Engineers Inc.,Vol. **2019**. pp. 9094–9100. doi:10.1109/ICRA.2019.8794221.

Qin, T., Li, P., and Shen, S. 2018. VINS-Mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans. Robot. **34**(4): 1004–1020.

Shan, M., Wang, F., Lin, F., Gao, Z., Tang, Y.Z., and Chen, B.M. 2015. Google map aided visual navigation for UAVs in GPS-denied environment.In 2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015. Institute of Electrical and Electronics Engineers Inc. pp. 114–119. doi:10.1109/ROBIO.2015.7418753.

Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., et al. 2021. InLoc: indoor visual localization with dense matching and view synthesis. IEEE Trans. Pattern Anal. Mach. Intell. **43**(4):1293–1307. doi:10.1109/TPAMI.2019.2952114.

Tateno, K., Tombari, F., Laina, I., and Navab, N. 2017. 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 2017, Vol. **2017**. Institute of Electrical and Electronics Engineers Inc., Janua, pp. 6565–6574. doi:10.1109/CVPR.2017.695.

Tennakoon, K.B., Silva, O.D., Jayasiri, A., Mann, G.K.I., and Gosine, R.G. 2021. Evaluation of a CNN-based visual place recognition system for GPS-denied navigation of VTOL vehicles. In Vertical Flight Society's 77th Annual Forum and Technology Display.

Valgren, C., and Lilienthal, A.J. 2007. Sift, surf and seasons: long-term outdoor localization using local features. In 3rd European Conference on Mobile Robots, ECMR'07, Freiburg, Germany, September 19–21, 2007. pp. 253–258.

Yan, X., Shi, Z., and Zhong, Y. 2018. Vision-based global localization of unmanned aerial vehicles with street view images. In Chinese Control Conference, CCC, Vol. **2018**. IEEE Computer Society, pp. 4672–4678. doi:10.23919/ChiCC.2018.8483081.

Yin, P., Xu, L., Li, X., Yin, C., Li, Y., Srivatsan, R.A., et al. 2019. A multi-domain feature learning method for visual place recognition. In Proceedings of the IEEE International Conference on Robotics and Automation, Vol. **2019**. Institute of Electrical and Electronics Engineers Inc. pp. 319–324. doi:10.1109/ICRA.2019.8793752.

Yu, L., Joly, C., Bresson, G., and Moutarde, F. 2016. Monocular urban localization using street view. In 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE. pp. 1–6.

Yu, L., Joly, C., Bresson, G., and Moutarde, F. 2017. Monocular urban localization using street view. In 2016 14th International Conference on Control, Automation, Robotics and Vision, ICARCV 2016. Institute

of Electrical and Electronics Engineers Inc. doi:10.1109/ICARCV.2016.7838744.

Yu, C., Liu, Z., Liu, X.J., Qiao, F., Wang, Y., Xie, F., et al. 2019. A DenseNet feature-based loop closure method for visual SLAM system. In IEEE International Conference on Robotics and Biomimetics, ROBIO 2019. Institute of Electrical and Electronics Engineers Inc. pp. 258–265. doi:10.1109/ROBIO49542.2019.8961714.

Zhang, Z. 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11): 1330–1334. doi:10.1109/34.888718.

Zhou, X., Meng, B., Dong, Y., Huang, X., and Zhang, K. 2021. An efficient image-based indoor positioning approach using ORB and LSH. In Proceedings of the 2021 China Automation Congress, CAC 2021. Institute of Electrical and Electronics Engineers Inc. pp. 2985–2989. doi:10.1109/CAC53003.2021.9727606.

Zhu, M., and Huang, L. 2021. Fast and robust visual loop closure detection with convolutional neural network. In 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer, ICFTIC 2021. Institute of Electrical and Electronics Engineers Inc. pp. 595–598. doi:10.1109/ICFTIC54370.2021.9647341.