Received: 18 June 2021 | Revised: 27 October 2021 | Accepted: 3 January 2022

DOI: 10.1002/rob.22060

RESEARCH ARTICLE

WILEY

Fast and incremental loop closure detection with deep features and proximity graphs

Shan An[1] | Haogang Zhu[1] | Dong Wei[2] | Konstantinos A. Tsintotas[3] | Antonios Gasteratos[3]

[1]State Key Lab of Software Development Environment, Beihang University, Beijing, China

[2]Tech & Data Center, JD.COM Inc., Beijing, China

[3]Department of Production and Management Engineering, Democritus University of Thrace, Xanthi, Greece

Correspondence
Haogang Zhu, State Key Lab of Software Development Environment, Beihang University, 100191 Beijing, China.
Email: haogangzhu@buaa.edu.cn

Funding information
Ministry of Science and Technology of the People's Republic of China

Abstract

In recent years, the robotics community has extensively examined methods concerning the place recognition task within the scope of simultaneous localization and mapping applications. This article proposes an appearance-based loop closure detection pipeline named "Fast and Incremental Loop closure Detection (FILD++). First, the system is fed by consecutive images and, via passing them twice through a single convolutional neural network, global and local deep features are extracted. Subsequently, a hierarchical navigable small-world graph incrementally constructs a visual database representing the robot's traversed path based on the computed global features. Finally, a query image, grabbed each time step, is set to retrieve similar locations on the traversed route. An image-to-image pairing follows, which exploits local features to evaluate the spatial information. Thus, in the proposed article, we propose a single network for global and local feature extraction in contrast to our previous work (FILD), while an exhaustive search for the verification process is adopted over the generated deep local features avoiding the utilization of hash codes. Exhaustive experiments on eleven publicly available data sets exhibit the system's high performance (achieving the highest recall score on eight of them) and low execution times (22.05 ms on average in New College, which is the largest one containing 52,480 images) compared to other state-of-the-art approaches.

KEYWORDS

learned-based features, loop closure detection, mapping, navigable small-world graph indexing, visual-based navigation

1 | INTRODUCTION

Autonomous robots have to explore unknown areas while retaining the capability to construct a reliable map of the environment (Garcia-Fidalgo & Ortiz, 2015; Kostavelis & Gasteratos, 2015). This process is widely known as simultaneous localization and mapping (SLAM) and constitutes an essential component for any modern robotic system (Cadena et al., 2016).

Besides, place recognition—the ability to match a scene with a different one located about the same spot—is necessary to generate a valid map (Lowry et al., 2016). In recent years, the mobile robot platforms' increased computational power allowed cameras to be established as the primary sensor to perceive the appearance of a scene (Cummins & Newman, 2008, 2011; Engel et al., 2015; Tsintotas et al., 2019). However, the noisy sensor measurements, modeling inaccuracies, and errors due to field abnormalities affect the performance of SLAM. Identifying known locations in the traversed route based on camera information to rectify the incremental pose drift is widely known as visual loop closure detection (Botterill et al., 2011; Han et al., 2021; Mei et al., 2010; Tsintotas, Bampis, &

Gasteratos, 2018; Zhang, 2011). This operation is highly related to image retrieval, as the system tries to find the most similar visual entry within a visual database, which is explicitly built using camera measurements gathered along a trajectory. There are two main stages in this process, namely *filtering* and *re-ranking* (Teichmann et al., 2019). Regarding *filtering*, the database elements are sorted according to their similarity to the query image, that is, the current robot's view. Then, during *re-ranking*, each candidate image-pair generated from the *filtering* is verified based on its spatial correspondences (Radenovic et al., 2018).

Early studies in image retrieval used global description vectors, such as color or texture, to represent the visual data (Konstantinidis et al., 2005; Oliva & Torralba, 2001, 2006; Torralba et al., 2003). The subsequent pipelines utilized the shape and local information extracted through point-of-interest detection and description methods to find the most similar candidates (Amanatiadis et al., 2011; Bay et al., 2006; Calonder et al., 2010; Lowe, 2004; Rublee et al., 2011). These approaches provided robust detection against rotation and scale changes. However, the increased time needed to extract and match local features constitutes a significant bottleneck, particularly in highly textured environments (Tsintotas et al., 2019). Therefore, researchers adopt more sophisticated solutions to overcome this drawback, such as quantizing the descriptor space, producing more compact representations, and faster indexing.

The so-called Bag-of-Words (BoW) model (Sivic & Zisserman, 2003), usually constructed through k-means clustering (MacQueen, 1967), employs the widely utilized term-frequency inverse-document-frequency (TF-IDF) technique to generate visual words histograms that represent the camera data. In many BoW-based place recognition approaches, the proper image-pair is retrieved via histogram comparisons (Bampis et al., 2016, 2018; Gálvez-López & Tardós, 2012; Mur-Artal & Tardós, 2014; Tsintotas, Bampis, Rallis, et al., 2018). Such methods exhibit high accuracy and low execution times, which are achieved due to the utilization of indexing techniques, for example, the hierarchical k-means tree (Nicosevici & Garcia, 2012), k-d tree (Liu & Zhang, 2012), and k-NN graph (Hajebi & Zhang, 2014). Nevertheless, their functionality is highly dependent on the training environment wherein the visual data are extracted and, in turn, on the produced vocabulary. Some visual loop closure detection frameworks incorporate mechanisms to map the environment through an incrementally generated visual vocabulary to cope with such dependencies (Angeli et al., 2008; Filliat, 2007; Khan & Wollherr, 2015; Labbe & Michaud, 2013; Tsintotas et al., 2018). However, due to their database construction, these pipelines mainly adopt voting techniques to indicate the most similar location within the traversed route.

Compared to hand-crafted, the features extracted from specific layers of convolutional neural networks (CNNs) show high discrimination power (An et al., 2019; Babenko et al., 2014; Gordo et al., 2016; Hou et al., 2015; Sünderhauf et al., 2015). Thus, CNN-extracted elements became a popular choice for many image classification (Krizhevsky et al., 2012) and scene recognition (Zhou et al., 2014) applications. Afterward, the proper image is selected through comparison techniques similar to BoW schemes. However, the spatial information embedded in image frames, which is crucial for data association between image-pairs for SLAM, is missing in the location's global representation. Hence, methods for extracting local features have been developed. DEep Local Feature (DELF), one of the original methods proposed for local CNN-based feature extraction, selects key-points based on an attention mechanism (Noh et al., 2017). Subsequently, the description stage is achieved using dense, localized features. Finally, principal component analysis (PCA) whitening reduces the descriptor space and improves the retrieval accuracy (Jégou & Chum, 2012).

In our previous work (An et al., 2019), CNN-based global features extracted by MobileNetV2 (Sandler et al., 2018) were fed into a Hierarchical Navigable Small World (HNSW) graph (Malkov & Yashunin, 2018) to map the environment via an incrementally generated visual database. In addition, the graph allowed for short indexing times when searching for Nearest Neighbors (NN). Then, local features, extracted via speeded up robust features (SURF) (Bay et al., 2006), were converted to binary codes to achieve real-time geometrical verification between the chosen image-pair. In this study, a similar scheme for mapping the robot's traversed path has been adopted. Besides, we utilize two forward passes through a single network for global and local features extraction. The main advantages offered by this strategy are: (i) the highly reduced execution time for feature extraction and (ii) a significant accuracy improvement due to CNN-based features' better representation. Furthermore, in this article, we introduce a *re-ranking* optimization, which is based on local features' low dimensional space (40 bins). Due to this fact, an exhaustive search is employed, unlike our previous work where hash codes were employed (Cheng et al., 2014), to improve the verification process. Finally, the proposed framework is more compact, simpler, and much faster than FILD (An et al., 2019). The presented algorithm is evaluated experimentally against a total of eleven benchmark data sets. As a final note, the source code[1] of our Fast and Incremental Loop closure Detection (FILD) pipeline, dubbed as "FILD++," is made publicly available to facilitate future studies.

The remainder of the paper is organized as follows: In Section 2, a literature review of the most prominent works on visual loop closure detection is given. Section 3 describes our deep features, and Section 4 introduces our HNSW visual database. In Section 5, the proposed detection pipeline is detailed, while the experimental protocol and the outcoming comparative results follow in Sections 6 and 7, respectively. Finally, Section 8 discusses the proposed approach, draws conclusions, and provides our plans.

2 | RELATED WORK

This section presents a literature review regarding the approaches which tackle the task of appearance-based loop closure detection. Depending on their visual feature extraction techniques, these pipelines are distinguished into two categories: hand-crafted features and CNN-based features.

[1] https://github.com/AnshanTJU/FILD

2.1 | Approaches using hand-crafted features

Since many researchers quantize the extracted features to generate a visual vocabulary and cope with the large amount of features, off-line and incremental approaches are presented according to the process they follow to construct their database. Fast appearance-based MAPping (FAB-MAP) is considered to be the most popular off-line approach (Cummins & Newman, 2008, 2011). It uses a pretrained SURF dictionary and a Chow Liu tree to learn its words' covisibility (Chow & Liu, 1968). BoWSLAM allows robots to navigate in unknown environments by utilizing the BoW feature matching with FAST corner detector and image patch descriptor (Botterill et al., 2011). Gálvez-López and Tardós (2012) proposed a hierarchical BoW model, built with local binary features in addition to direct and inverse indexes. Their method was improved by employing ORB features (Rublee et al., 2011) to incorporate rotation and scale invariance properties (Mur-Artal & Tardós, 2014). Similarly, previously visited locations were detected inside a Parallel Tracking and Mapping (PTAM) framework (Klein & Murray, 2007). Bampis et al. (2016, 2018) combined the visual words' occurrences of sequence segments, that is, groups-of-images, to assist the matching process. Recently, points and lines were combined based on information entropy to realize accurate loop closure detection (Han et al., 2021).

While the approaches mentioned above relied on a static visual vocabulary adapted to the training environment, in the work of Angeli et al. (2008) an incrementally constructed vocabulary was proposed. Loops were identified via the matching probability of a Bayesian scheme. In a similar manner, an agglomerative clustering algorithm was adopted for database generation (Nicosevici & Garcia, 2012). The stability between visual elements' associations was attained using an incremental image-indexing process in conjunction with a tree-based feature-labeling method. Real-Time Appearance-Based Mapping (RTAB-Map) proposed a memory management mechanism to limit the number of candidate locations (Labbe & Michaud, 2013). An Incremental bag of BInary words for Appearance-based Loop closure Detection (IBuILD) was proposed by Khan & Wollherr (2015). Visual words were generated via feature matching on consecutive images, while a likelihood function decided about the location pairing. Hierarchical Topological Mapping (HTMap) proposed by Garcia-Fidalgo and Ortiz (2017) relied on a loop closure scheme based on the Pyramid Histogram of Oriented Gradients (PHOG) (Bosch et al., 2007). Similar locations are highlighted due to binary local features' correspondences. An incremental approach exerting binary descriptors and dynamic islands was proposed in the work of Garcia-Fidalgo and Ortiz (2018), while Tsintotas et al. (2018) dynamically segmented the incoming image stream to formulate places represented by unique visual words. A probabilistic voting scheme followed, aiming to indicate the proper place, while an image-to-image pairing was held based on the locations' spatial correspondences. The same authors, proposed a mapping algorithm based on an incrementally generated visual vocabulary constructed through local features tracking (Tsintotas et al., 2019). The authors improved their method through the addition of a temporal filter and a vocabulary management technique in Tsintotas et al. (2021). The candidate locations were chosen through their probabilistic binomial score (Gehrig et al., 2017). A modified growing self-organizing network was proposed by Kazmi and Mertsching (2019) for learning the topological representation of global gist features (Oliva & Torralba, 2001).

2.2 | Approaches using CNNs features

The impressive performance of CNNs, exhibited on a wide variety of tasks, has been the main reason for their becoming the principal solution to many visual place recognition systems. Utilizing an end-to-end trainable and generalized VLAD layer (Jégou et al., 2010), NetVLAD was proposed for similar locations' identification (Arandjelovic et al., 2016). A Spatial Pyramid-Enhanced VLAD (SPE-VLAD) layer was proposed by Yu et al. (2020) to encode the feature extraction and improve the loss function. PCANet (Chan et al., 2015) employed a cascaded deep network to extract unsupervised features improving the loop closure detection pipeline (Xia et al., 2016). Cascianelli et al. (2017) proposed a visual scene modeling technique that preserved the geometric and semantic structure and, at the same time, improved the appearance invariance. A multiscale pooling exertion allowed for condition- and viewpoint-invariant features to be generated (Chen et al., 2017). Omnidirectional CNN was proposed to mitigate the challenge of extreme camera pose variations (Wang et al., 2018). In the work of Chen et al. (2018), the authors proposed an attention mechanism capable of being incorporated into an existing feed-forward network architecture to learn image representations for long-term place recognition. A useful similarity measurement for detecting revisited locations in changing environments was proposed by Xin et al. (2019). The combination of a neural network inspired by the Drosophila olfactory neural circuit with an 1D Continuous Attractor Neural Network resulted into a compact system exhibiting high performance (Chancán et al., 2020). Such works commonly used CNNs to extract the global descriptor of a scene, while few of them applied CNNs to extract local information for appearance-based loop closure detection.

3 | IMAGE REPRESENTATION

Our feature extraction module relies on a fully convolutional network (FCN) to generate specific representations from the incoming image frames. Aiming to achieve an enhanced image representation, the proposed architecture is implemented upon a modified version of DELF employing both global and local features in different scales through a double-pass process. We choose the initial three convolutional blocks of ResNet50 (He et al., 2016) as the backbone of our network, while the output of the last layer is fed into the feature extraction module as depicted in Figure 1. Aiming for compact and discriminative features, image-level labeled information is used to train the network.

3.1 | Incoming image frame

To extract representative and robust features, we use different scales to extract the global and local features. Contrary to the original version of

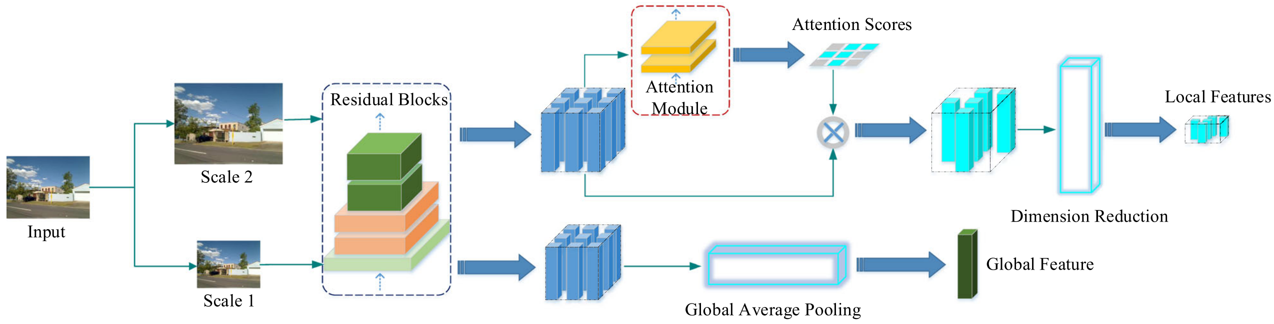


FIGURE 1 The modified version of DEep Local Feature (DELF) (Noh et al., 2017) architecture for feature extraction. We extract the incoming visual stream's global and local representations via two passes of the proposed fully convolutional network. Three components constitute its structure, namely: the backbone, the global feature extraction branch, and the local feature extraction branch. The first component is based on the residual blocks of ResNet50 (He et al., 2016). Simultaneously, for the global and local branches, an average pooling technique, an attention module, and a dimension reduction method are used. The attention module is adopted for the corresponding scores' generation. This way, the most relevant features are assigned with higher scores before their dimensionality reduction. More specifically, the first-scale features are fed to the global branch, while the features from the second are sent to the local branch [Color figure can be viewed at wileyonlinelibrary.com]

DELF, where image pyramids are constructed using seven different scales, the proposed system employs only two of them; one for global and the other for local features' extraction. According to the experiments, the scales for extracting global and local features are set as 0.5 and 1.4. Our method performs even better when specific scales for different data sets are used. However, the forward processing on multiple scales requires more times than a single scale. Due to this fact, the selected parameters (see Section 7.2) provide a good trade-off between accuracy and timing, affording a system with low complexity and high performance.

3.2 | The backbone network

Our backbone network comprises the first three convolutional blocks of ResNet50. The initial block contains a 7×7 convolution layer followed by a Batch Normalization (BN), a Rectified Linear Unit (ReLU), and a 3×3 max-pooling layer with stride 2. The second convolutional block includes three residual blocks each of which comprises three layers: 1×1, 3×3, and 1×1, respectively. The 1×1 convolution layers are used to reduce/increase the feature map's channels, while a BN and a ReLU follow each layer. The final convolutional block comprises four residual blocks, which are similar to the previous block but are considered for the feature map's dual channels.

3.3 | Global features

A global average pooling (GAP) layer (Lin et al., 2013) is applied to the output feature map $w \times h \times c$ of the backbone network to produce a single description vector for the incoming visual data. Here, w, h, c are feature map's width, height, and channels, respectively. As a result, the feature map's dimensionality is reduced to $1 \times 1 \times c$ since GAP generates a single number per channel, which is the average of all $w \times h$ values. GAP's output forms the employed global feature.

3.4 | Local features

3.4.1 | Attention-based local features

Each pixel in the backbone network's output is considered as a local grid; the feature map is the dense sample of this grid. The tensor composed of all grid channels is treated as a local feature, while the corresponding keypoint is located at the center of the receptive field in the pixel coordinates.

Since not all the densely extracted elements are appropriate for the intended recognition task, an attention module consisting of two 1×1 convolutional layers is applied to select a subset of them. This module aims to learn a score function for each local feature and creates the corresponding score map of size $w \times h \times 1$. A softplus activation (Dugas et al., 2001) is deployed in the second layer to ensure the score is nonnegative. Then, the elements which present a value higher than a score threshold are selected. It is noted that in this case, the local features are first computed and then selected. This process differs from the hand-crafted techniques wherein the keypoints are firstly detected, and then their description vectors are generated.

The score map learning process is the same as the original version. The features to be learned by the attention model are denoted as $f_n \in \mathbb{R}^d$, $n = 1, \ldots, N$, with d is the feature dimension. The score function for each feature is $\alpha(f_n; \theta)$, with θ denoted the paramter of function $\alpha(\cdot)$. The network generates the output logit y by a weighted sum of the feature vectors:

$$y = \mathbf{W}\left(\sum_n \alpha(f_n; \theta) \cdot f_n\right), \tag{1}$$

$\mathbf{W} \in \mathbb{R}^{M \times d}$ is the weight of the final fully connected layer of the network. M is the number of classes to be predicted.

The cross-entropy loss is used for the training, which is defined as:

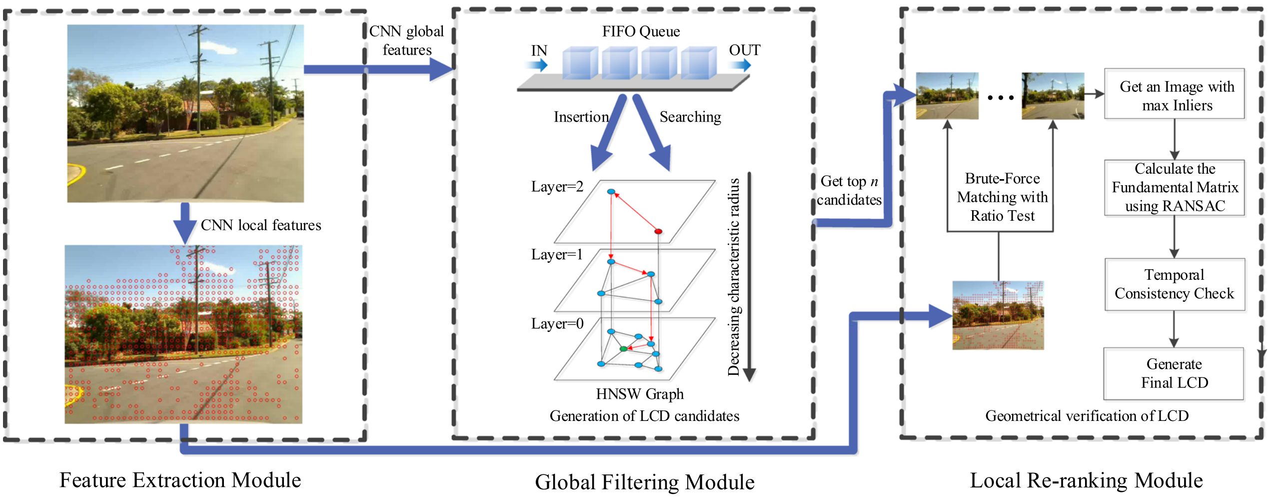


FIGURE 2 An overview of the proposed loop closure detection pipeline. Global and local convolution neural network (CNN)-based features are extracted as the incoming image stream enters the system. The global features enter the first-in-first-out (FIFO) queue, and subsequently, they are fed into the HNSW graph (Malkov & Yashunin, 2018), to generate the incremental database. Simultaneously, the top n nearest neighbors are retrieved using the global feature, while a brute-force matching technique between the candidate image-pairs is performed at the local features space. A ratio test is implemented to eliminate false matches in conjunction with a RANSAC-based geometrical verification check. Finally, a temporal consistency check is employed to approve the final loop closure pair [Color figure can be viewed at wileyonlinelibrary.com]

$$\mathcal{L} = -y^* \cdot \log\left(\frac{\exp(y)}{\mathbf{1}^T \exp(y)}\right). \tag{2}$$

Here y^* denotes ground-truth in one-hot representation. $\mathbf{1}$ is one vector. The backpropagation is used to train the parameters $\alpha(\cdot)$. The gradient is defined as:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial y}\sum_n \frac{\partial y}{\partial \alpha_n}\frac{\partial \alpha_n}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial y}\sum_n \mathrm{W}\mathrm{f}_n \frac{\partial \alpha_n}{\partial \theta}. \tag{3}$$

3.4.2 | Local features' dimensionality reduction

A commonly used feature dimension reduction method (Jégou & Chum, 2012) is incorporated to reduce the dimension of local features. We first preprocess the local features with L2 normalization. Then, their dimension is reduced using PCA to generate 40-dimensional features. Finally, the features are processed again through an L2 normalization, as it has been demonstrated by Jégou and Chum (2012) that the renormalization provides a better mean average precision in image retrieval tasks.

4 | HNSW GRAPH DATABASE

Our system employs the HNSW graph to index the generated global features. The proposed method is selected as it constitutes a reliable technique that outperforms other contemporary approaches, such as tree-based BoW (Muja & Lowe, 2014), product quantization (Jegou et al., 2011), and locality sensitive hashing (Andoni & Razenshteyn, 2015). The following subsections describe its properties and the way HNSW is used to construct the graph-based visual database.

4.1 | Hierarchical navigable small world

HNSW is a fully graph-based incremental k-nearest neighbor search (k-NNS) structure, as shown in Figure 2. It is based on the Navigable Small World (NSW) model (Kleinberg, 2000), which follows a logarithmic or polylogarithmic scaling of greedy graph routing. Such models are important for understanding the underlying mechanisms of real-life networks' formation.

A graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ formally consists of a set of nodes (i.e., feature vectors) $\boldsymbol{V}$ and a set of links $\boldsymbol{E}$ between them. A link e_{ab} connects node a with node b in a directional manner, that is, form a to b, on HNSW. The neighborhood of a is defined as the set of its immediately connected nodes. HNSW exploits strategies for explicit selection of the graph's enter-point node, separates links of different length scales, and chooses neighbors using an advanced heuristic. Then, the search process is performed in a hierarchical multilayer graph, which allows logarithmic scalability.

4.2 | Database construction

In BoW-based approaches, the visual vocabulary is usually constructed using k-means clustering. A search index is built over the visual words, which are generated using feature descriptors extracted from a training data set.

HNSW has the property of incremental graph building (Malkov & Yashunin, 2018). The image features can be consecutively inserted into the graph structure. An integer maximum layer l is randomly selected with an exponentially decaying probability distribution for every inserted element. The insertion process starts from the top layer to the next layer by greedily traversing the graph to find the ef closest neighbors to the inserted element q in the layer. The found closest neighbors from the previous layer will be used as an enter point to the next layer. A greedy search algorithm is used to find the closest neighbors in each layer. The process repeats until the connections of the inserted elements are established on the zero layer. In each layer higher than zero, the maximum number of connections that an element can have per layer is defined by the parameter M, which is the only meaningful construction parameter. The construction process of the HNSW graph is illustrated in the middle of Figure 2.

During the mobile robot's movement, the deep global features of the images are inserted into the graph vocabulary. The whole process is on-line and incremental, thus eliminating the need for prebuilt vocabulary. Therefore, the use of HNSW ensures the robot's working in various environments.

4.3 | *k*-NN search

The k-NN search algorithm in HNSW is roughly equivalent to the insertion algorithm for an item in layer $l = 0$. The difference is that the closest neighbors found at the ground layer are returned as the search result. The search quality is controlled by the parameter ef.

The distance between two global features or nodes in the HNSW graph, indicates the corresponding images' similarity. We use the normalized scalar product (cosine of the angle between vectors) to compute the distance of two nodes during graph construction and k-NN search, which is calculated as follows:

$$s_{pq} = \frac{X_p^T \cdot X_q}{\|X_p\|_2 \cdot \|X_q\|_2}. \tag{4}$$

where s_{pq} is the distance score between images I_p, I_q and X_p, X_q are the global description vectors. $\|X\|_2 = \sqrt{X^T X}$ denotes the Euclidean norm of vector X. Since we aim to build a computational inexpensive system, we have chosen to make use of the advanced vector extensions instructions to accelerate the distance computation.

5 | DETECTION PIPELINE

5.1 | System overview

As the robotic platform navigates into the working area, its incoming sensory information, provided by the mounted camera, passes through the CNN to extract the visual features. First, the global features enter the first-in-first-out (FIFO) queue, aiming to avoid early visited locations' detection, and then are placed into the database. The n most similar locations, indicated by k-NN, are selected, while an image-to-image correlation eliminates false-positive matches through a ratio test. Eventually, geometrical and temporal consistency checks are employed to generate the final loop closure pair. An overview of the proposed scheme is illustrated in Figure 2, while its steps are described in Algorithm 5.1.

Algorithm 1 Our loop closure detection pipeline

Input: the image I_i captured by the visual sensory module during robot's navigation; the excluded area, defined by frame N_{non} as $\psi \times \phi$, where ψ is a temporal constant and ϕ is the frame rate of the camera; the returned number of nearest neighbors n; the threshold of inlier points τ.

Input: whether the i detection constitute a loop closure or not.

initialize a First In First Out (FIFO) queue Q. **While** true **do** ▷ perform the loop closure detection pipeline during robot's mission.
  $I_i \leftarrow$ read the current image.
  X_i, $L_i \leftarrow$ extract global and local visual features.
  If ($i > N_{non}$) **then**
    $X_{pre} \leftarrow$ pop the FIFO queue Q.
    add X_{pre} to HNSW graph visual database.
    k-nearest neighbor search of X_i in the database to obtain the n among them.
    $inlier_{max} \leftarrow -1$, $ind \leftarrow -1$
    for r = 1 to n **do**
      perform geometrical verification for L_i and L_r
      If failed **then**
        continue
      end if
      $inlier \leftarrow$ the number of inliers
      If $inlier > inlier_{max}$ **then**
        $inlier_{max} \leftarrow inlier$
        $ind \leftarrow r$
      end if
    end for
    temporal consistency check for L_i and L_{ind}
    If success **then**
      Loop detected.
    end if
  end if
  push X_i to the FIFO queue Q.
end while

5.2 | Retrieval strategy

The n most similar locations are determined via the HNSW's k-NN search using the query's extracted global feature. Since the image frames are captured sequentially, the adjacent locations to query, that is, images acquired in close time proximity, are highly possible to share semantic information yielding to high similarities among them. When searching the database this area should be avoided, so as to keep the system safe from false-positive detections. Therefore, we use the FIFO queue to store images' global representations. As shown in Algorithm 5.1, the global feature X_i, belonging to image I_i, firstly enters the queue Q, and subsequently it remains there aiming to be inserted at the HNSW graph when the robot runs out of the non-search area. The non-search area is defined based on a temporal constant ψ, and the camera's frame rate ϕ. Consequently, when we use the current feature as query, it will only search in database area defined via $N - \psi \times \phi$, where N is the number of the entire set of camera measurements up to time i. As a final note, the images in the non-search area will never appear in the results.

5.3 | Image-to-image matching

As described in Section 3, we extract local deep features for each incoming image. Thus, the matching process is performed between the query q and the n closest neighbors based on a brute-force matching algorithm. This technique is rarely reported in the literature for visual data association due to the presented high complexity. However, when low-dimensional floating-point global descriptors are used, such in our case, brute-force matching does not demand relatively high computation time. At last, a distance ratio check (Lowe, 2004), defined through a threshold ε, is employed on the proposed pair.

5.4 | Geometrical verification

Our system incorporates a geometrical verification step to discard outliers, that is, false-positive detections. To achieve this, we compute the fundamental matrix T between the chosen candidate pair of images using a RANdom SAmple Consensus (RANSAC)-based scheme (Torr & Murray, 1997). We record the candidate with the highest number of inliers when the calculation succeeds.

5.5 | Temporal consistency check

As a final step, a temporal consistency check is employed intending to examine whether the aforementioned conditions are met for β consecutive camera measurements similarly to Tsintotas et al. (2018). This way, the proposed pipeline may lose a possible loop closing identification in cases where the query image is the initial in a sequence of pre-visited locations, however, we prefer to prevent the system from wrong the identifications preserving. When the aforementioned conditions are met, the matched pair is recognized as a loop closure event.

6 | BENCHMARK

6.1 | Benchmark data sets

Eleven challenging and publicly available image-sequences have been chosen to evaluate the performance of our framework. These data sets are captured in different operating environments, for example, various lighting conditions, strong visual repetition, and dynamic occlusions such as cars and pedestrians. A detailed description of each image-sequence is listed in Table 1. Regarding KITTI vision suite (Geiger et al., 2012), Malaga 2009 Parking 6L (Malaga6L) (Blanco et al., 2009), and St. Lucia (Glover et al., 2010), the incoming visual stream is obtained via a camera mounted on a moving car, while New College (Smith et al., 2009), and City Center (Cummins & Newman, 2008) are recorded through the vision system of a wheeled robot. Malaga6L, New College and City Center are composed of stereo images; in the course of our experiments, we have chosen the left camera stream for the first and the right camera for the rest. Our experimental setup is chosen according to the work of Kazmi and Mertsching (2019).

6.2 | Ground truth labeling

Commonly, the ground truth data, referring to the correct loop events, is generated according to the global positioning system (GPS) logs. For example, St. Lucia and Malaga6L utilize a GPS distance-range of 10 and 4 m from the query, respectively, to define the ground truth. We carefully checked this data for each data set recognizing that some image pairs were not accurately labeled, as shown in Figure 3. In many cases, this occurs owing to the robot traversing through locations that surpass the GPS's distance threshold, though the captured visual content might be similar. However, in such cases, if a valid fundamental matrix is computed, the transformation matrix between the two camera poses can be available. Such pairs should be treated as true-positive loop closure events. Another problem concerns the situation wherein the robot's viewpoint differs from the viewpoint confronted in its first traversal. Regardless of the system being precisely located at the same place, these image pairs are considered true negative events. An exemplar case of this situation is illustrated in Figure 4.

Considering the GPS logs are not accurate, we adopt human labeling for the ground truth generation. We produce image pairs which are located less than 40 m in GPS logs. Then these pairs are labeled by asking whether they are from the same place by crowdsourcing. During labeling, if a decision was hard to be taken, the

TABLE 1 Descriptions of the used data sets

Data set		Description	Image resolution (px)	# images	Frame rate (Hz)	Distance (km)
KITTI (Geiger et al., 2012)	Seq# 00	Outdoor, dynamic	1241 × 376	4541	10	3.7
	Seq# 02		1241 × 376	4661		5.0
	Seq# 05		1226 × 370	2761		2.2
	Seq# 06		1226 × 370	1101		1.2
Oxford	New College (Smith et al., 2009)	Outdoor, dynamic	512 × 384	52480	20	2.2
	City Center (Cummins & Newman, 2008)		640 × 480	1237	10	1.9
Malaga 2009 (Blanco et al., 2009)	Parking 6L	Outdoor, slightly dynamic	1024 × 768	3474	7	1.2
St. Lucia (Glover et al., 2010)	100909 (12:10)	Outdoor, dynamic	640 × 480	19,251	15	~17.6
	100909 (14:10)			20,894		
	180809 (15:45)			21,434		
	190809 (08:45)			21,815		

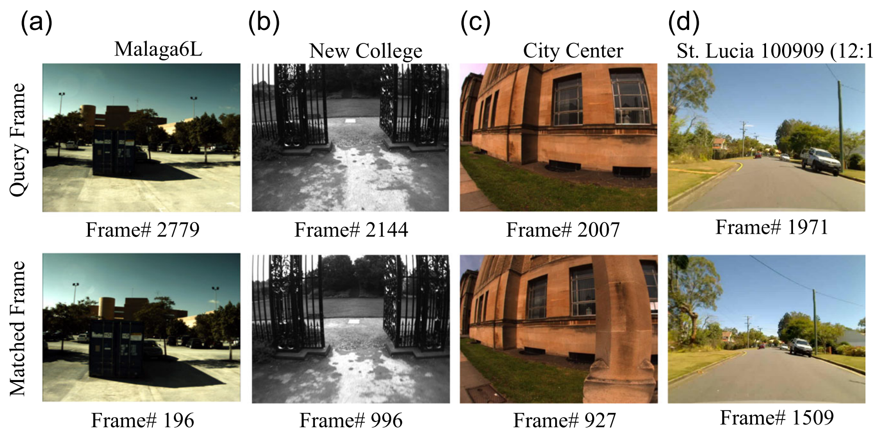


FIGURE 3 Examples of image pairs which are not correctly labeled in the ground truth data derived via the global positioning system logs. (a) Malaga6L, (b) New College, (c) City Center, and (d) St. Lucia 100909 (12:10)

proposed pairs are rechecked by experts familiar with the place recognition task. Each of the aforementioned data sets is processed two times before used, while for the KITTI image-sequences, the data were accurate enough avoiding this procedure. Our accurate, manually labeled ground truth files are made publicly available to facilitate further studies.

7 | EXPERIMENTAL RESULTS

This section presents the experiments conducted to demonstrate the proposed pipeline's effectiveness. Our setup including training strategy, parameters and evaluation metrics are introduced in Section 7.1, while different settings for the proposed features' extraction module are evaluated in Section 7.2. Next, we analyze the HNSW parameterization in Section 7.3, and evaluate the geometrical verification process in Section 7.4. A comparison of our global feature with two contemporary CNN-based features is presented in Section 7.5. The system's performance and quantitative comparison with the state-of-the-art are presented in Section 7.6. Finally, we measure our system's complexity on the representative data sets in Section 7.7.

7.1 | Experimental settings

7.1.1 | Training strategy

Since our feature extractor is hard to get trained directly, owing to the employment of the attention module, a two-step strategy is applied. First, our base network is trained, leaving the attention module out; subsequently, two fully connected layers (FCLs) are adjoined for the classification. ResNet50, upon which the proposed system is

FIGURE 4 A labeling correction: the image sequence in the first row shows the robot's trajectory as it turns to the right road, while in the second row, it turns to the left road at the same place. Frames #835 and #147 are visually different but are labeled as loops according to the GPS, for its distance is lower than 10 m. During our experiments, these images are considered as true negative pairs

TABLE 2 The recall at 100% precision and the feature extraction speed (ms) on different scales of global and local features

	Scales (Local)													
	0.25		0.35		0.5		0.7		1.0		1.4		2.0	
Scales (Global)	recall	speed	recall	speed	recall	speed	recall	speed	recall	speed	recall	speed	recall	speed
0.25	0.9123	8.11	0.9073	8.90	0.9010	10.01	0.9123	12.54	0.9135	16.49	0.9261	28.25	0.9236	56.07
0.35	0.8972	8.73	0.9273	9.40	0.9110	10.56	0.9110	13.19	0.8960	17.14	0.8960	28.86	0.9023	56.28
0.5	0.8972	9.61	0.9110	10.28	0.9098	11.23	0.9261	14.00	0.9110	17.93	0.9492	29.62	0.9480	57.15
0.7	0.8972	11.73	0.9110	12.41	0.8997	13.56	0.9248	16.09	0.9098	19.90	0.9492	31.79	0.9492	59.01
1.0	0.8910	14.61	0.8985	15.25	0.8935	16.22	0.9261	18.77	0.9035	22.64	0.9211	34.53	0.9323	61.90
1.4	0.8922	20.95	0.9035	21.69	0.8935	22.47	0.9286	24.96	0.9048	28.89	0.9223	40.26	0.9336	68.97
2.0	0.8947	35.08	0.9023	35.57	0.8960	36.46	0.9261	38.95	0.9060	42.76	0.9223	53.97	0.9336	81.00

built, is trained on the ImageNet (Russakovsky et al., 2015) and then the model is fine-tuned on a large-scale landmark data set (Weyand et al., 2020). The cross-entropy loss is used for the image classification.

Next, when the base network is trained, its weights are squeezed. The attention module is added, and the resulted score map is used to pool the features by a weighted sum. Subsequently, the features enter the fully connected layer for the classification with the cross-entropy loss. Finally, we use this model to obtain discriminative deep features.

7.1.2 | Training parameters

Our network was trained through the stochastic gradient descend (SGD) optimizer. An initial learning rate of 0.001 and 25 epochs as the maximum number for training was selected, with its rate being halved every 10 epochs. Similarly, the same optimizer was chosen for the attention module with an initial learning rate set at 0.01 at the maximum number of 20 epochs, while the learning rate is halved every 10 epochs. We implemented the two networks using the batch size of 256.

TABLE 3 Parameters list

Image scale for global feature extraction s_g	0.5
Image scale for local feature extraction s_l	1.4
Score threshold of local feature δ	15
Number of nearest to q elements to return, ef	40
Maximum number of connections for each element per layer, M	48
Search area time constant, ψ	40
Ratio of ratio test, ε	0.7
Images temporal consistency, β	2
Number of of matches for geometrical verification, n	5

7.1.3 | Baseline approaches

The compared methods include classic and recently published place recognition systems namely: DLoopDetector (Gálvez-López & Tardós, 2012), Tsintotas et al. (2018), PREVIeW (Bampis

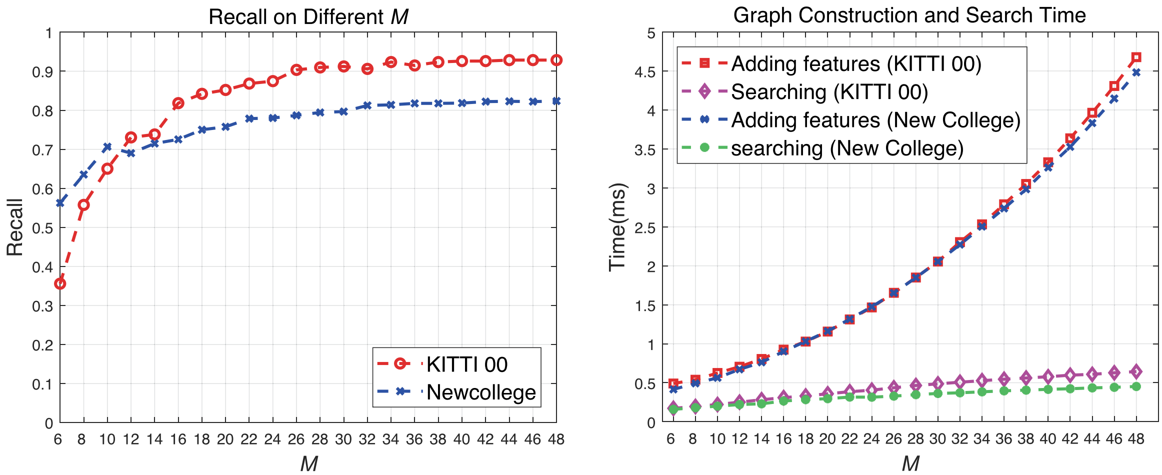


FIGURE 5 Evaluating the parameter M on KITTI 00 (Geiger et al., 2012) and New College (Smith et al., 2009). (Left) Our pipeline's recall scores for perfect precision using a variety of values ranging from 6 to 48. (Right) The timing needed for new feature addition and database search

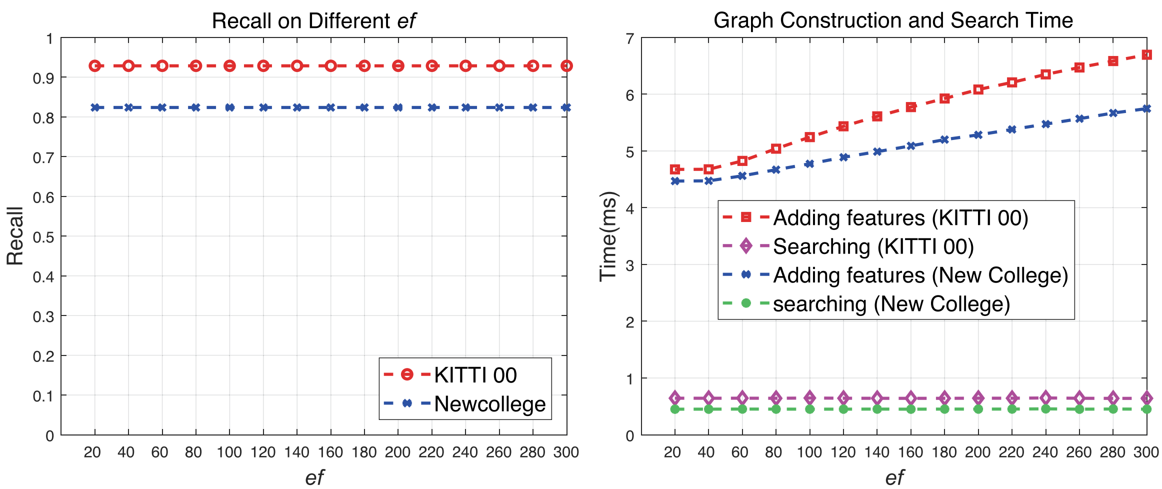


FIGURE 6 Evaluating the parameter ef on KITTI 00 (Geiger et al., 2012) and New College (Smith et al., 2009). (Left) Our pipeline's recall scores for perfect precision using a variety of values ranging from 20 to 300. (Right) The timing needed for new feature addition and database search

et al., 2018), iBoW-LCD (Garcia-Fidalgo & Ortiz, 2018), Kazmi et al. (Kazmi & Mertsching, 2019), as well as our previous method (An et al., 2019). Most of the chosen methods are implemented using the respective open-source codes. For Kazmi's method, we directly report their results as published in their article.

7.1.4 | Evaluation metrics

For the loop closure detection task, the commonly used metric is the recall rate at 100% precision. The precision-recall metric is defined as:

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}, \tag{5}$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}, \tag{6}$$

where true-positives is the number of correct identifications, indicating the detected loop closures are true loops according to the ground truth. False-positives is the number of wrong detections, representing the identifications found by the algorithm; however, these are not labeled to ground truth. False-negatives indicate the number of true loop closure events, which are not found by the algorithm.

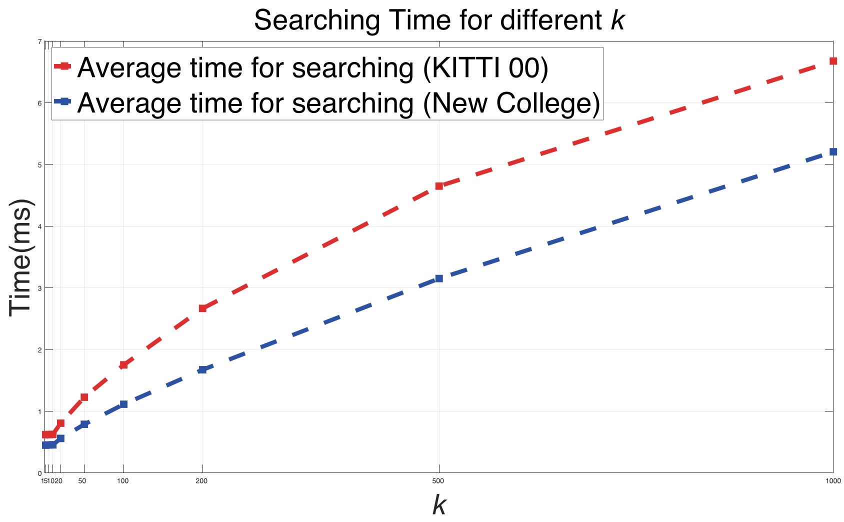


FIGURE 7 The searching time for different k on the KITTI 00 data set (Geiger et al., 2012) and the New College data set (Smith et al., 2009)

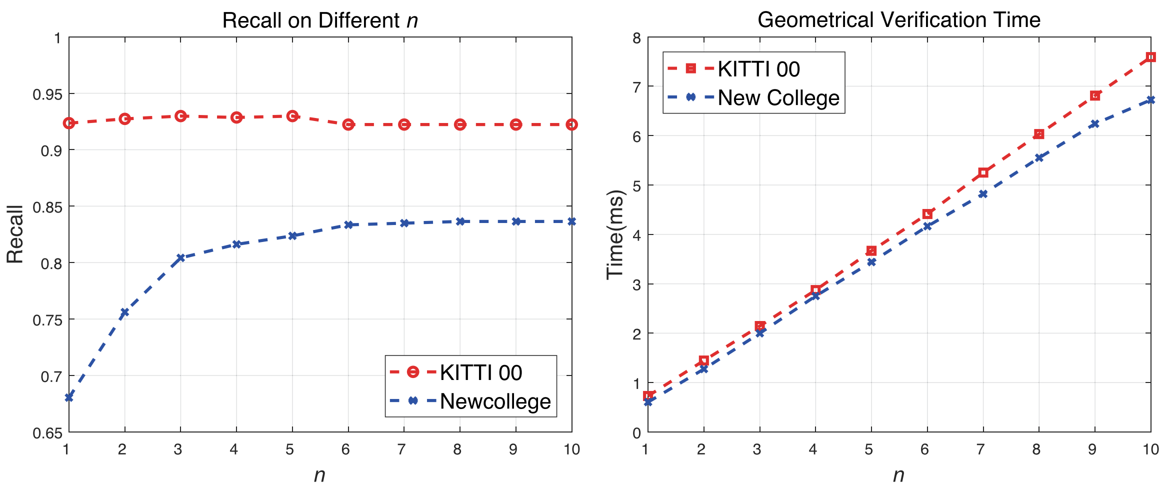


FIGURE 8 Evaluating the parameter n on KITTI 00 (Geiger et al., 2012) and New College (Smith et al., 2009). (Left) Our pipeline's recall scores for 100% precision using a variety of values n. (Right) The timing needed for geometrical verification

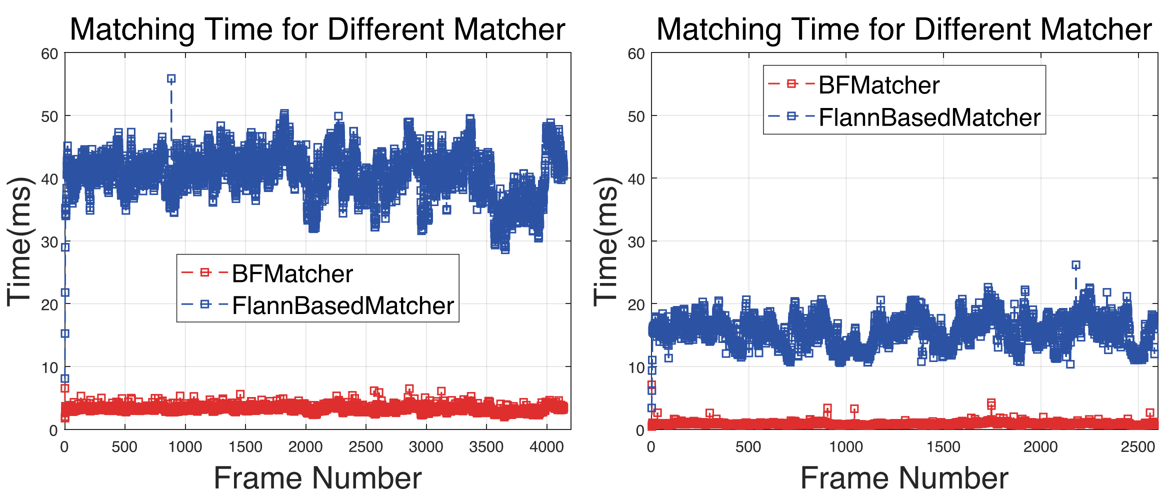


FIGURE 9 The image matching time of our algorithm on the KITTI 00 data set (Geiger et al., 2012) (Left) and the New College data set (Smith et al., 2009) (Right) using different matching strategies

TABLE 4 Recalls at 100% precision: a comparison of our method with different CNN-based global features

Data set		NetVLAD (Arandjelovic et al., 2016)	Resnet50-AP-GeM (Revaud et al., 2019)	FILD++
KITTI	Seq# 00	91.88	91.24	**94.92**
	Seq# 02	**74.77**	73.21	73.52
	Seq# 05	91.81	94.70	**95.42**
	Seq# 06	**98.90**	97.79	98.16
Oxford	New College	83.35	**84.85**	82.37
	City Center	89.84	**90.56**	90.01
Malaga 2009	Parking 6L	59.83	60.11	**62.74**
St. Lucia	100909 (12:10)	80.46	79.26	**83.39**
	100909 (14:10)	63.80	58.10	**66.41**
	180809 (15:45)	79.67	69.36	**81.36**
	190809 (08:45)	83.21	82.91	**87.86**

Note: Bold values denotes the best.

TABLE 5 Average feature extraction time (ms) comparison of our method with different CNN-based global features

Methods	KITTI 00	City Center	Malaga6L	St. Lucia 100909 (12:10)
NetVLAD (Arandjelovic et al., 2016)	105.60	94.25	131.07	85.97
Resnet50-AP-GeM (Revaud et al., 2019)	22.60	19.90	35.33	16.93
Proposed Global Feature	**11.23**	**8.38**	**17.25**	**8.54**

Note: Bold values denotes the best.

TABLE 6 Recalls at 100% precision: a comparison of the baseline methods with our framework

Data set		DLoopDetector (Gálvez-López & Tardós, 2012)[a]	Tsintotas et al. (Tsintotas et al., 2018)	PREVIeW (Bampis et al., 2018)[b]	iBoW-LCD (Garcia-Fidalgo & Ortiz, 2018)[c]	Kazmi et al. (Kazmi & Mertsching,2019)[d]	FILD (An et al., 2019)	FILD++
KITTI	Seq# 00	72.43	93.18	89.47	76.50	90.39	91.23	**94.92**
	Seq# 02	68.22	76.01	71.96	72.22	**79.49**	65.11	73.52
	Seq# 05	51.97	94.20	87.71	53.07	81.41	85.15	**95.42**
	Seq# 06	89.71	86.03	80.15	95.53	97.39	93.38	**98.16**
Oxford	New College	47.56	52.44	80.87	73.14	51.09	76.74	**82.37**
	City Center	30.59	16.30	49.63	82.03	75.58	66.48	**90.01**
Malaga 2009	Parking 6L	31.02	59.14	33.93	57.48	50.98	56.09	**62.74**
St. Lucia	100909 (12:10)	37.22	26.27	60.93	70.02	80.06	76.06	**83.39**
	100909 (14:10)	14.87	9.77	23.06	**68.06**	58.10	53.84	66.41
	180809 (15:45)	31.36	15.07	49.79	**87.50**	72.55	66.96	81.36
	190809 (08:45)	39.78	27.68	56.69	59.36	80.13	78.00	**87.86**

[a]Compared to Gálvez-López and Tardós (2012), we use different number of images for New College and Malaga6L. We have changed the normalized similarity score threshold to achieve 100% precision, as there are false detections using the default parameters.

[b]We report the recall using the default parameters. However, the precision of each data set cannot achieve 100%.

[c]We report the iBoW-LCD recalls on KITTI data set from Kazmi and Mertsching (2019).

[d]We quote the results as reported in Kazmi and Mertsching (2019), as an open-source implementation were not available.

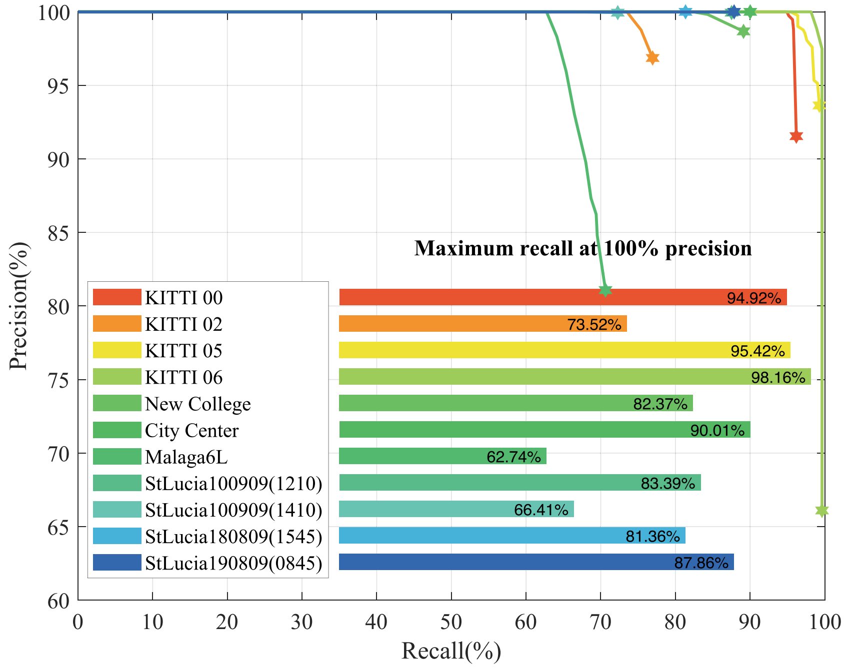


FIGURE 10 Our algorithm's precision-recall curves on each evaluated data set

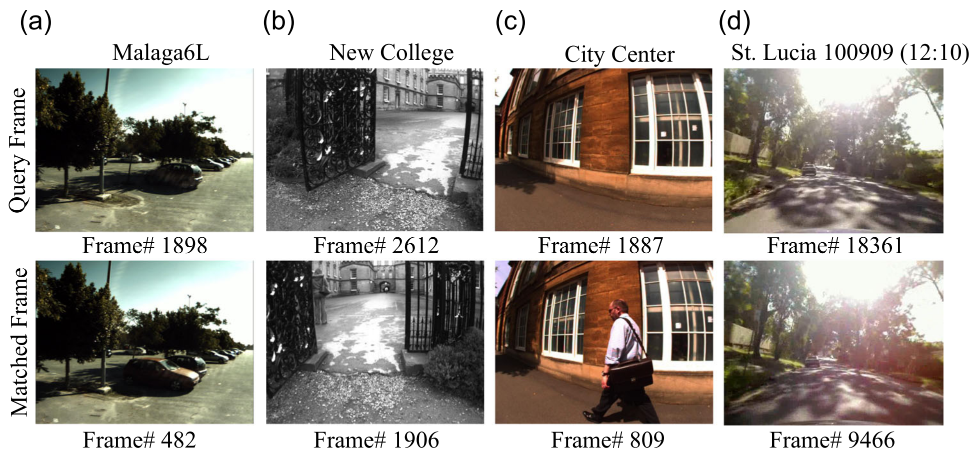


FIGURE 11 Some example images of the detected loop-closure locations. (a) Malaga6L, (b) New College, (c) City Center, and (d) St. Lucia 100909 (12:10)

7.1.5 | Implementation

Experiments were performed on a Linux machine with an Intel Xeon CPU E5-2640 v3 (2.60 GHz) and an NVIDIA Tesla P40 GPU. More specifically, only feature extraction was performed on the GPU; any other operation ran on the CPU. The proposed network is implemented via TensorFlow, yet bindings are provided in C++. Besides, to test the speed on embedded devices, we additionally implemented FILD++ on an NVIDIA Jetson TX2 GPU and report the respective outcome in Section 7.7.

7.2 | Image scales evaluation

The original DELF utilizes image pyramids to generate descriptors of different scales. It uses seven different scales ranging from 0.25 to 2.0, which are a $\sqrt{2}$ factor apart. As processing times are crucial for mobile robotic applications, we propose to use only one scale for global feature extraction and another one scale for local feature extraction.

We conduct extensive experiments to evaluate the recall and the extraction speed of using different feature extraction scales. For KITTI 00 data set, the results of different combination of scales for extracting global and local features are given in Table 2. The three

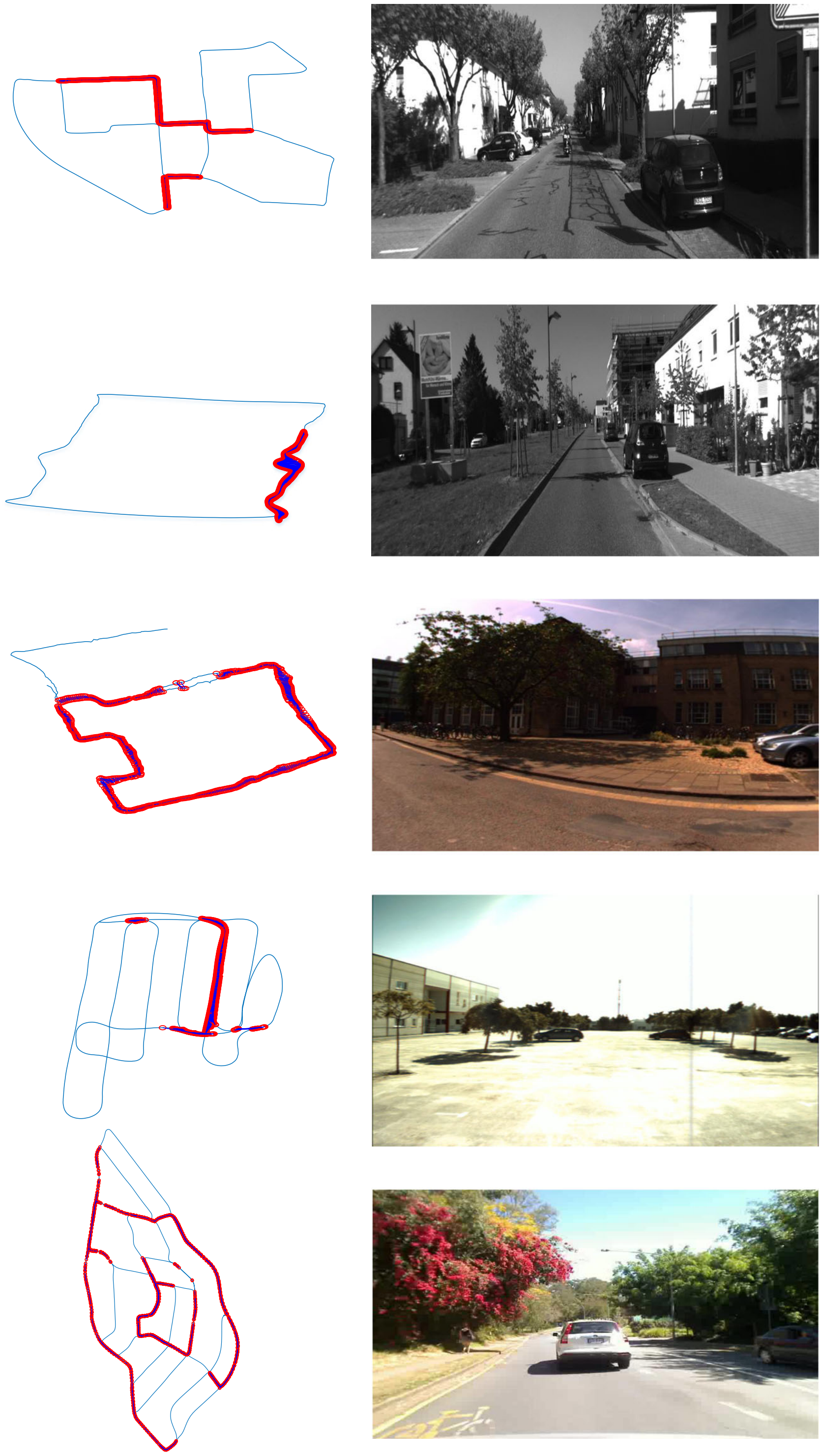


FIGURE 12 Robot trajectories (left) and example images (right). From top to bottom: KITTI 00 (Geiger et al., 2012), KITTI 06 (Geiger et al., 2012), City Center (Cummins & Newman, 2008), Malaga6L (Blanco et al., 2009), St. Lucia 100909 (12:10) (Glover et al., 2010). The loop closure detections are labeled using red circles

highest recall scores are marked in blue. As shown, the scales of 0.7 and 2.0 for global and local features, respectively, reach the highest recall rate at 100% precision. A similar score is obtained at the scales of 0.5 and 0.7 for global features and 1.4 for local features. However, considering the extraction time, we chose the scales of 0.5 and 1.4 which achieve the same recall through a timing below 30 ms. It is also notable that for the scales of 0.25 for both global and local deep features, the extraction time is only 8.11 ms. Our algorithm's parameters are summarized in Table 3, determined via the experimentation reported in Sections 7.2, 7.3, and 7.4.

7.3 | HNSW parameters' evaluation

For HNSW graph construction and searching, there are two parameters that could affect the search quality: the number of

TABLE 7 Average execution time (ms/query) on the representative data sets

Approach	KITTI 00	City Center	Malaga6L	St. Lucia 100909 (12:10)
DLoopDetector (Gálvez-López & Tardós, 2012)	111.04	27.51	42.57	91.04
Tsintotas et al. (Tsintotas et al., 2018)	521.54	183.23	638.61	625.05
PREVIeW (Bampis et al., 2018)	32.39	34.09	36.33	25.40
FILD (An et al., 2019)	62.68	40.23	68.16	49.10
FILD++	38.70	32.10	56.56	34.20

TABLE 8 Average execution time (ms/query) of our method in different data sets

Stages	KITTI 00	City Center	Malaga6L	St. Lucia 100909 (12:10)
Feature extraction	29.67	22.04	45.41	22.48
Adding feature	4.69	4.30	3.63	6.03
Graph searching	0.64	0.33	0.47	0.70
Feature matching	3.32	2.17	4.13	2.26
RANSAC	0.38	3.26	2.92	2.73
Whole system	38.70	32.10	56.56	34.20

nearest to q elements to return, ef; and the maximum number of connections for each element per layer, M. The range of the parameter ef should be within 300, because the increase in ef will lead to little extra performance but in exchange, significantly longer construction time. The range of the parameter M should be 5–48 (Malkov & Yashunin, 2018). The experiments in Malkov and Yashunin (2018) show that a bigger M is better for high recall and high dimensional data, which also defines the memory consumption of the algorithm.

We perform the experiments on the KITTI 00 and the New College data sets to choose M and ef for the HNSW graph. The parameter ef is set to 40 when we change M. The number of matches for geometrical verification n is set to 5. As 100% precision can be reached with the temporal consistency check. The recalls are shown in the left part of Figure 5. We can see when M increases, the recall will also increase. In the right part of Figure 5, the feature adding time and searching time will be increased when M increases. To achieve a better recall, we choose $M = 48$ in the following experiments.

For evaluating different ef, it can be seen that in the left part of Figure 6, the recall does not significantly change when the ef increases. In the right part of Figure 6, the feature adding time will be increased when ef increases, while the searching time remains with no growth. Therefore, $ef = 20$ was selected.

Besides, we evaluate the searching time of the HNSW graph for different returned number k of nearest neighbors. As shown in Figure 7, we can see that the searching method costs nearly logarithmic time when increase the returned nearest neighbors. The time cost accords with the time complexity of the HNSW graph (Malkov & Yashunin, 2018).

7.4 | Evaluating geometrical verification

As image-to-image matching through RANSAC is computationally costly, we evaluate the parameter n using values ranging from 1 to 10. As shown in Figure 8, the timing needed for geometrical verification increases linearly with n, as a new RANSAC estimation needs to be done on each round. For KITTI 00, timing varies from 0.73 to 7.59 ms for $n = [1, 2, 3, ..., 10]$. New College timing varies from 0.61 to 6.72 ms. However, the higher the value of n the better the performance. Aiming to achieve a trade-off between recall and computational complexity, we have chosen $n = 5$. We empirically fix the ratio test ε to 0.7. This value is frequently used for image matching using SIFT (Lowe, 2004) and SURF (Bay et al., 2006).

Furthermore, we evaluate the processing time for two different image matching strategies namely: the FLANN matcher (Muja & Lowe, 2009) and brute-force matcher. As shown in Figure 9, the brute-force matcher's timing is significantly lower than FLANN. For KITTI 00, the average score is 3.32 ms, while for FLANN is 40.70 ms. Respectively, for New College, the timings are 0.91 and 15.62 ms. This happens due to the proposed local features' low dimension (40-dimensional).

7.5 | Evaluating deep global features

A comparison of our global feature against two other contemporary CNN-based features is presented. NetVLAD (Arandjelovic et al., 2016) and Resnet50-AP-GeM (Revaud et al., 2019) have been selected since these are the features commonly used as feature extractors in place recognition. For our experiments, features extracted from NetVLAD and Resnet50-AP-GeM replaced our global representations. However, the other modules remain the same. As we can see in Table 4, the features provided by FILD++ achieve the highest recall rate in most of the evaluated data sets. We also recorded the timing needed for feature extraction when different

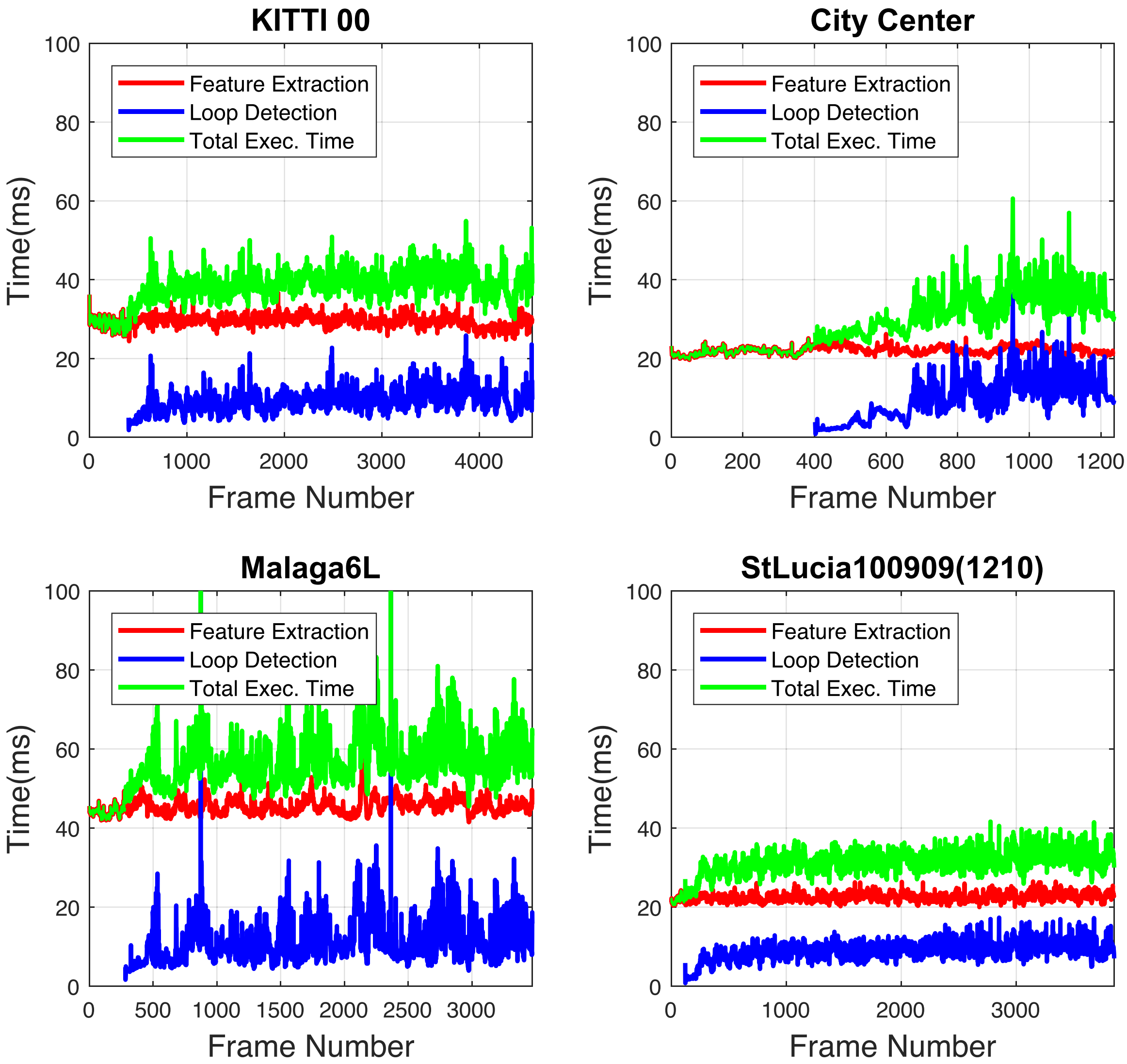


FIGURE 13 Execution times of our algorithm

TABLE 9 Average execution time (ms/query) in New College with 52480 images

Stages	Mean	SD	Max	Min
Feature extraction	14.62	0.65	21.13	12.30
Adding feature	3.97	2.47	22.63	0.04
Graph searching	0.67	0.19	3.20	0.04
Feature matching	1.06	0.10	14.04	0.08
RANSAC	1.72	1.08	17.01	0.0
Whole system	22.05	5.04	58.98	14.56

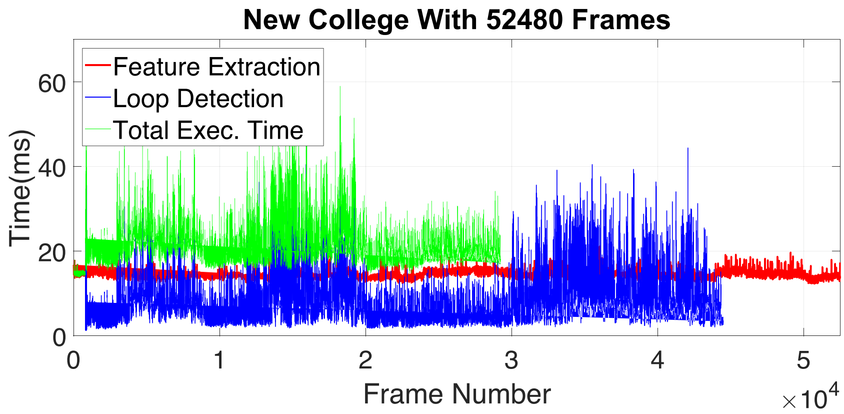
extractors are used in Table 5. Our method requires only 11.23 ms when applied on City Center, while NetVLAD and Resnet50-AP-GeM need 105.60 and 22.60 ms, respectively. The results show that our global feature outperforms the other methods in terms of speed and recall.

7.6 | FILD++ performance

In Table 6, we list our system's highest recall score at 100% precision on 11 data sets, while compared to the baseline methods. As shown FILD++ outperforms the other methods on 8 out of 11 image-sequences. Malaga6L is recorded at a parking site, thus presents high scene similarity due to the absence of distinct differences between the roads. Therefore, each of the evaluated methods performs poorly. As far as the KITTI vision suite and Oxford data sets are concerned, improved performance is demonstrated. This is mostly owing to architectural constructions appearing in these environments, which are similar to the training set of our feature extraction network. Hence, our pipeline can extract more representative deep features and compare them more precisely for these types of scenes.

Figure 10 illustrates the precision-recall curves generated by varying the number of RANSAC inliers. Our framework can successfully detect loops through a recall score ranging from 62.74% (Malaga6L) to 98.16% (KITTI 06). Malaga6L is the most

FIGURE 14 Execution times in New College with 52,480 images



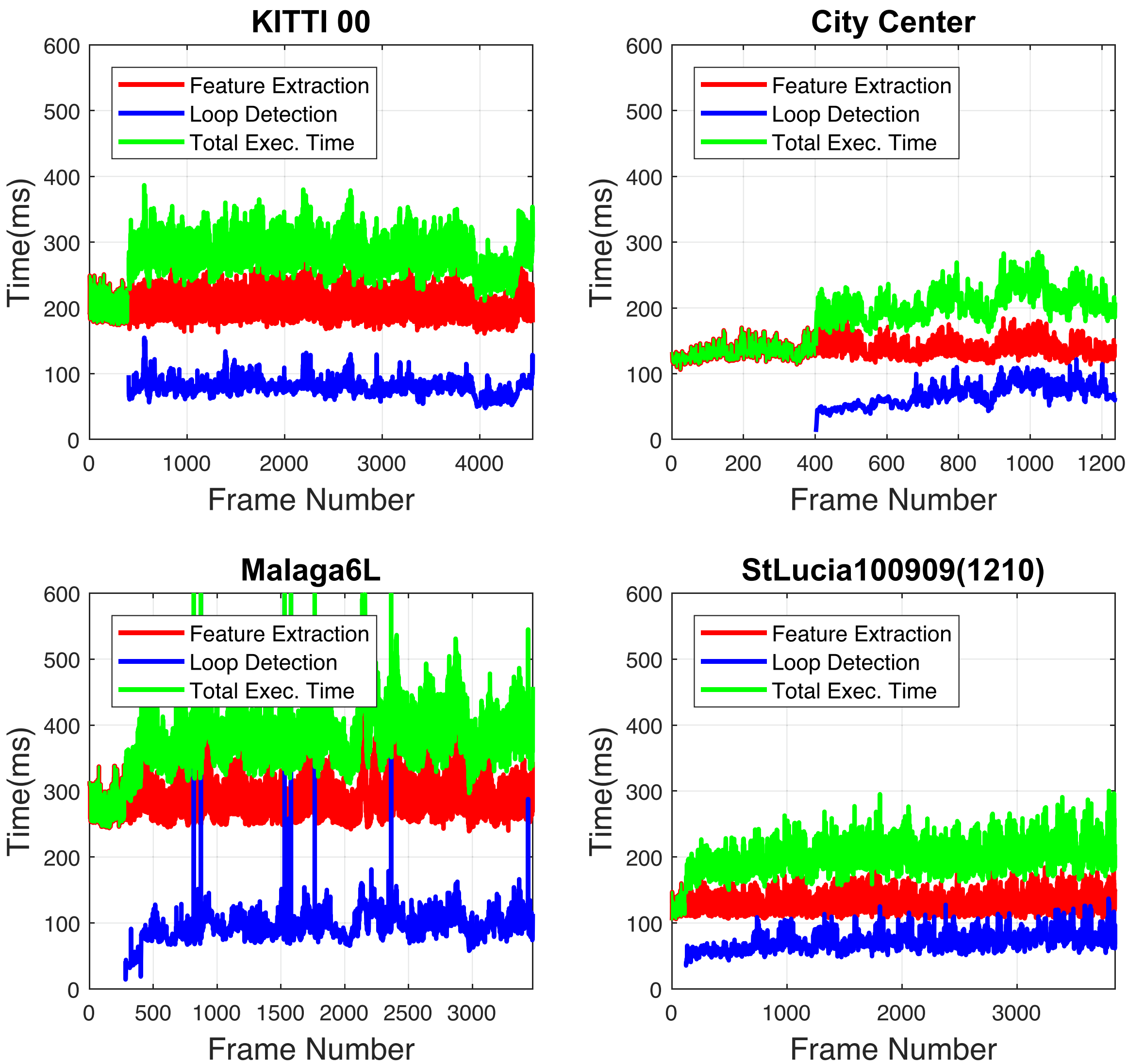


FIGURE 15 Execution times on an NVIDIA Jetson TX2 GPU

challenging data set and KITTI 06 is the smallest data set among the rest. Some examples of TP detections are shown in Figure 11. It is worth noting that when dynamic objects are included, for example, cars in Figure 11a and people in Figure 11c, FILD++ can correctly identify the previsited location. The example in Figure 11b demonstrate that our system can handle the viewpoint changes, while Figure 11d shows its ability to deal with illumination variations. We show the loop closure detections detected by our framework on to of the robot's trajectories in Figure 12.

TABLE 10 Average Execution Time (ms/query) on an NVIDIA Jetson TX2 GPU

Stages	KITTI 00	City Center	Malaga6L	St. Lucia 100909 (12:10)
Feature extraction	200.12	135.97	292.06	128.19
Loop detection	78.84	68.44	96.60	72.39
Whole system	278.96	204.41	388.66	200.58

TABLE 11 Average execution time (ms/query) of FILD (An et al., 2019) and FILD++ in New College with 52,480 images

Method	FILD (An et al., 2019)	FILD++
Feature extraction	17.69	14.62
Hash codes creation	16.94	0.0
Adding feature	5.21	3.97
Graph searching	0.93	0.67
Feature matching	2.23	1.06
RANSAC	7.55	1.72
Whole system	50.28	22.05

7.7 | Time requirements

We have estimated our system's complexity on four representative data sets. As shown in Table 7, FILD++ achieves a higher speed than its predecessor. In general, this improvement is owing to the local features' low dimensionality which permits faster image matching.

The average execution times for different pipeline stages are presented in Table 8. Also, in Figure 13, we present the timings for features' extraction and loops' detection as a function of frame number. As illustrated, FILD++ requires constant time for each data set, while the features' extraction is the most costly procedure. For Malaga6L, our pipeline needs about 50 to 80 ms for the total execution time, while the feature extraction requires about 45 ms. This happens due to the images' resolution, which is the largest among the evaluated data sets. Concurrently, for St. Lucia, the average timing is below 40 ms, because of the different image resolution. Furthermore, it is observed that the timing for our indexing graph-based technique is below 1 ms and the whole system's speed ranges from 32 to 57 ms demonstrating FILD++'s high efficiency. In Table 9, we test our system's scalability setting the frequency of New College to f = 20 Hz and obtained 52,480 images. The average execution time is about 22 ms. As can be seen in Figure 14, an increase of frames number would not induce a rise of processing time.

We also implemented our algorithm on the Jetson TX2 platform in Max-N mode (all CPU cores in use and GPU clocked at 1.3 GHz) and show the timing in Figure 15. FILD++ does not require extra processing time even if applied in an embedded platform. The most time-consuming stage is the features' extraction as we perform two forward passes for each image frame. In Table 10, we list the average time for the feature extraction, the loop detection and the whole system. The proposed system processes Malaga6L in 388.66 ms, while for any other data set, processing times are below 300 ms indicating its low computational complexity.

8 | DISCUSSION AND CONCLUSION

In this article, a visual loop closure detection approach is proposed, dubbed as FILD++. Through two forward passes of a single network, our system extracts global and local deep features for filtering and reranking, respectively. Along with the robot's navigation, an HNSW graph is built incrementally based on the global features permitting fast indexing and database search during query. When a candidate location is retrieved it is geometrically verified using the provided local features. Eleven publicly available data sets are chosen for our evaluation showing FILD++'s effectiveness and efficiency compared with other state-of-the-art approaches.

The proposed FILD++ framework has three advantages compared with the previous FILD method. First, the proposed framework is more compact. This is because only one network was used for feature extraction. In addition, the extracted deep local features are only 40-dimensional, which is significantly lower than SURF (128-dimensional). Because there is only one network and without the usage of CasHash (Cheng et al., 2014), the source code of FILD++ is more concise than FILD, as given in the GitHub[2]. Besides, the dimension of global feature in FILD++ is also lower than that in FILD, which is 1024-dimensional *versus* 1280-dimensional.

Second, the proposed method is simpler than the previous method. For feature extraction, FILD extracts global features using MobileNetV2 and local features using SURF. We simplify the feature extraction process in this study. The deep global features and local features are extracted via two forward passes of a single network. This dramatically simplifies the feature extraction process. Because the dimension of the deep local feature extracted by our method is only 40-dimensional, we can use a brute-force matcher for efficient feature matching. Therefore we did not use CasHash (Cheng et al., 2014) in FILD++. As a result, the hash code creation process is unnecessary, which simplifies the whole process.

Last but not least, FILD++ is much faster than its previous version. As shown in Table 7, FILD++ costs 38.70 ms per query on KITTI 00 data set, while FILD requires 62.68 ms. Thus, it can be seen that FILD++ is significantly faster than FILD on all data sets. Table 11 also shows the average execution time of FILD and FILD++ in the New College data set (52,480 images). As can be seen, the feature extraction in FILD needs more time than in FILD++. The hash codes creation step in FILD is also time-consuming, while there is no such step in FILD++. Because SURF features in FILD are different from the deep local features, we extracted in FILD++, the timing for RANSAC scheme is different. We can see our approach also takes less time at

[2]https://github.com/anshan-ar/FILD

this step. The overall time cost of the proposed FILD++ is 22.05 ms per query, while for FILD is 50.28 ms. This indicates the speed advantage of our new method when applied in large data sets.

Our system's performance depends on several factors: the reliability of its deep features, the HNSW's retrieval precision, and the effectiveness of the geometrical verification. The similarity scores of the query and the candidate images are not utilized. A proper threshold may have helped us with FP elimination; however, the complexity of the system would be inevitably high. During geometrical verification, as the number of matches n is an important parameter, the easiest way to achieve a higher recall is to increase its value. Howbeit, as illustrated, such action is time-demanding; therefore, a convenient trade-off is considered.

Our plans include the integration of the proposed method to a SLAM framework, while an increase of the classification accuracy will lead to higher performance. Consequently, using more powerful networks, such as ResNeXt (Xie et al., 2017) and ResNeSt (Zhang et al., 2020), we should be able to improve the system's performance.

ACKNOWLEDGMENTS

The authors wish to gratefully acknowledge Dr. Mark Cummins for his kindly help and Guangfu Che, whose constructive suggestions helped the system evaluation. This study was funded by grants from the National Key R&D Program of China (Grant No. 2021YFB2700300).

ORCID

Shan An http://orcid.org/0000-0001-7796-6952
Haogang Zhu http://orcid.org/0000-0003-1771-2752
Antonios Gasteratos http://orcid.org/0000-0002-5421-0332

REFERENCES

Amanatiadis, A., Kaburlasos, V., Gasteratos, A., & Papadakis, S. (2011). Evaluation of shape descriptors for shape-based image retrieval. *IET Image Processing*, *5*, 493–499.

An, S., Che, G., Zhou, F., Liu, X. L., Ma, X., & Chen, Y. (2019). Fast and incremental loop closure detection using proximity graphs. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 378–385).

Andoni, A., & Razenshteyn, I. (2015). Optimal data-dependent hashing for approximate near neighbors. In: *Proceedings of ACM Symposium of Theory of computing* (pp. 793–801).

Angeli, A., Filliat, D., Doncieux, S., & Meyer, J. (2008). Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, *24*, 1027–1037.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). Netvlad: CNN architecture for weakly supervised place recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5297–5307).

Babenko, A., Slesarev, A., Chigorin, A., & Lempitsky, V. (2014). Neural codes for image retrieval. In: *European Conference on Computer Vision* (pp. 584–599). Springer.

Bampis, L., Amanatiadis, A., & Gasteratos, A. (2016). Encoding the description of image sequences: A two-layered pipeline for loop closure detection. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4530–4536).

Bampis, L., Amanatiadis, A., & Gasteratos, A. (2018). Fast loop-closure detection using visual-word-vectors from image sequences. *International Journal of Robotics Research*, *37*, 62–82.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In: *European Conference on Computer Vision* (pp. 404–417).

Blanco, J.-L., Moreno, F.-A., & Gonzalez, J. (2009). A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, *27*, 327.

Bosch, A., Zisserman, A., & Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In: *Proceedings of ACM IntInternational Conference on Image and Video Retrieval* (pp. 401–408). ACM.

Botterill, T., Mills, S., & Green, R. (2011). Bag-of-words-driven, single-camera simultaneous localization and mapping. *Journal of Field Robotics*, *28*, 204–226.

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

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. In: *European Conference on Computer Vision* (pp. 778–792).

Cascianelli, S., Costante, G., Bellocchio, E., Valigi, P., Fravolini, M. L., & Ciarfuglia, T. A. (2017). Robust visual semi-semantic loop closure detection by a covisibility graph and cnn features. *Robotics and Autonomous Systems*, *92*, 53–65.

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2015). PCANet: A simple deep learning baseline for image classification? *IEEE Transactions of Image Processing*, *24*, 5017–5032.

Chancán, M., Hernandez-Nunez, L., Narendra, A., Barron, A. B., & Milford, M. (2020). A hybrid compact neural architecture for visual place recognition. *IEEE Robotics and Automation Letters*, *5*, 993–1000.

Chen, Z., Jacobson, A., Sünderhauf, N., Upcroft, B., Liu, L., Shen, C., Reid, I., & Milford, M. (2017). Deep learning features at scale for visual place recognition. In: *IEEE International Conference on Robotics and Automation* (pp. 3223–3230).

Chen, Z., Liu, L., Sa, I., Ge, Z., & Chli, M. (2018). Learning context flexible attention model for long-term visual place recognition. *IEEE Robotics and Automation Letters*, *3*, 4015–4022.

Cheng, J., Leng, C., Wu, J., Cui, H., & Lu, H. (2014). Fast and accurate image matching with cascade hashing for 3d reconstruction. In: *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–8).

Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, *14*, 462–467.

Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, *27*, 647–665.

Cummins, M., & Newman, P. (2011). Appearance-only SLAM at large scale with FAB-MAP 2.0. *International Journal of Robotics Research*, *30*(9), 1100–1123.

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., & Garcia, R. (2001). Incorporating second-order functional knowledge for better option pricing. In *Advances in Neural Information Processing Systems* (pp. 472–478).

Engel, J., Stückler, J., & Cremers, D. (2015). Large-scale direct slam with stereo cameras. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1935–1942).

Filliat, D. (2007). A visual bag of words method for interactive qualitative localization and mapping. In: *IEEE International Conference on Robotics and Automation* (pp. 3921–3926).

Gálvez-López, D., & Tardós, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, *28*, 1188–1197.

Garcia-Fidalgo, E., & Ortiz, A. (2015). Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, *64*, 1–20.

Garcia-Fidalgo, E., & Ortiz, A. (2017). Hierarchical place recognition for topological mapping. *IEEE Transactions on Robotics*, *33*, 1061–1074.

Garcia-Fidalgo, E., & Ortiz, A. (2018). iBoW-LCD: An appearance-based loop-closure detection approach using incremental bags of binary words. *IEEE Robotics and Automation Letters*, *3*, 3051–3057.

Gehrig, M., Stumm, E., Hinzmann, T., & Siegwart, R. (2017). Visual place recognition with probabilistic voting. In: *IEEE International Conference on Robotics and Automation* (pp. 3192–3199).

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In: *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3354–3361).

Glover, A. J., Maddern, W. P., Milford, M. J., & Wyeth, G. F. (2010). Fab-map. ratslam: Appearance-based slam for multiple times of day. In: *IEEE International Conference on Robotics and Automation* (pp. 3507–3512). IEEE.

Gordo, A., Almazán, J., Revaud, J., & Larlus, D. (2016). Deep image retrieval: Learning global representations for image search. In: *European Conference on Computer Vision* (pp. 241–257). Springer.

Hajebi, K., & Zhang, H. (2014). An efficient index for visual search in appearance-based slam. In: *IEEE International Conference on Robotics and Automation* (pp. 353–358). IEEE.

Han, J., Dong, R., & Kan, J. (2021). A novel loop closure detection method with the combination of points and lines based on information entropy. *Journal of Field Robotics*, *38*, 386–401.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).

Hou, Y., Zhang, H., & Zhou, S. (2015). Convolutional neural network-based image representation for visual loop closure detection. In: IEEE International Conference on Information and Automation, 2238–2245.

Jégou, H., & Chum, O. (2012). Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. In: *European Conference on Computer Vision* (pp. 774–787).

Jegou, H., Douze, M., & Schmid, C. (2011). Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*, 117–128.

Jégou, H., Douze, M., Schmid, C., & Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In: *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3304–3311). IEEE.

Kazmi, S. A. M., & Mertsching, B. (2019). Detecting the expectancy of a place using nearby context for appearance-based mapping. *IEEE Transactions on Robotics*, *35*, 1352–1366.

Khan, S., & Wollherr, D. (2015). IBuILD: Incremental bag of binary words for appearance based loop closure detection. In: *IEEE International Conference on Robotics and Automation* (pp. 5441–5447).

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In: *IEEE/ACM International Symposium on Mixed and Augmented Reality* (pp. 225–234).

Kleinberg, J. M. (2000). Navigation in a small world. *Nature*, *406*, 845.

Konstantinidis, K., Gasteratos, A., & Andreadis, I. (2005). Image retrieval based on fuzzy color histogram processing. *Optics Communications*, *248*, 375–386.

Kostavelis, I., & Gasteratos, A. (2015). Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, *66*, 86–103.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Labbe, M., & Michaud, F. (2013). Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, *29*, 734–745.

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. arXiv preprint arXiv:1312.4400.

Liu, Y., & Zhang, H. (2012). Indexing visual features: Real-time loop closure detection using a tree structure. In: *IEEE International Conference on Robotics and Automation* (pp. 3613–3618).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Compter Vision.*, *60*, 91–110.

Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., & Milford, M. J. (2016). Visual place recognition: A survey. *IEEE Transactions on Robotics 32*, 1–19.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In: *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281–297).

Malkov, Y. A., & Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. In: *IEEE Transactions in Pattern Analysis and Machine Intelligence*.

Mei, C., Sibley, G., & Newman, P. (2010). Closing loops without places. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3738–3744).

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In: *International Conference on Compter Vision Theory and Applications* (Vol. 2, p. 2).

Muja, M., & Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (pp. 2227–2240).

Mur-Artal, R., & Tardós, J. D. (2014). Fast relocalisation and loop closing in keyframe-based SLAM. In: *IEEE International Conference on Robotics and Automation* (pp. 846–853).

Nicosevici, T., & Garcia, R. (2012). Automatic visual bag-of-words for online robot navigation and mapping. *IEEE Transactions on Robotics*, *28*, 886–898.

Noh, H., Araujo, A., Sim, J., Weyand, T., & Han, B. (2017). Large-scale image retrieval with attentive deep local features. In: *IEEE Conference on Computer Vision* (pp. 3456–3465).

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*, 145–175.

Oliva, A., & Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, *155*, 23–36.

Radenovic, F., Iscen, A., Tolias, G., Avrithis, Y., & Chum, O. (2018). Revisiting oxford and paris: Large-scale image retrieval benchmarking. In: *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5706–5715).

Revaud, J., Almazán, J., Rezende, R. S., & Souza, C. R. d. (2019). Learning with average precision: Training image retrieval with a listwise loss. In: *IEEE Conference on Computer Vision* (pp. 5107–5116).

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In: *IEEE Conference on Computer Vision* (pp. 2564–2571).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Compter Vision*, *115*, 211–252.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In: *IEEE/IVF Conference on Computer Vision and Pattern Recognition* (pp. 4510–4520). IEEE.

Sivic, J., & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In: *IEEE Conference on Computer Vision* (p. 1470).

Smith, M., Baldwin, I., Churchill, W., Paul, R., & Newman, P. (2009). The new college vision and laser data set. *International Journal of Robotics Research*, *28*, 595–599.

Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., & Milford, M. (2015). On the performance of convnet features for place recognition. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4297–4304).

Teichmann, M., Araujo, A., Zhu, M., & Sim, J. (2019). Detect-to-retrieve: Efficient regional aggregation for image search. In: *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5109–5118).

Torr, P. H., & Murray, D. W. (1997). The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, *24*, 271–300.

Torralba, A., Murphy, K. P., Freeman, W. T., & Rubin, M. A. (2003). Context-based vision system for place and object recognition. In: *IEEE Conference on Computer Vision* (Vol. 3, pp. 273–280).

Tsintotas, K. A., Bampis, L., & Gasteratos, A. (2018a). Assigning visual words to places for loop closure detection. In: *IEEE International Conference on Robotics and Automation* (pp. 5979–5985).

Tsintotas, K. A., Bampis, L., & Gasteratos, A. (2018b). DOSeqSLAM: Dynamic on-line sequence based loop closure detection algorithm for SLAM. In: *IEEE International Conference on Imaging Systems and Techniques* (pp. 1–6).

Tsintotas, K. A., Bampis, L., & Gasteratos, A. (2019). Probabilistic appearance-based place recognition through bag of tracked words. *IEEE Robotics and Automation Letters*, *4*, 1737–1744.

Tsintotas, K. A., Bampis, L., & Gasteratos, A. (2021). Modest-vocabulary loop-closure detection with incremental bag of tracked words. *Robotics and Autonomous Systems*, *141*, 103782.

Tsintotas, K. A., Bampis, L., Rallis, S., & Gasteratos, A. (2018). SeqSLAM with bag of visual words for appearance based loop closure detection. In: *Proceedings of International Conference on Robotics in Alpe-Adria Danube Region* (pp. 580–587).

Tsintotas, K. A., Giannis, P., Bampis, L., & Gasteratos, A. (2019). Appearance-based loop closure detection with scale-restrictive visual features. In: *Proceedings of International Conference on Computer Vision Systems* (pp. 75–87).

Wang, T.-H., Huang, H.-J., Lin, J.-T., Hu, C.-W., Zeng, K.-H., & Sun, M. (2018). Omnidirectional cnn for visual place recognition and navigation. In: *IEEE International Conference on Robotics and Automation* (pp. 2341–2348).

Weyand, T., Araujo, A., Cao, B., & Sim, J. (2020). Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2575–2584.

Xia, Y., Li, J., Qi, L., & Fan, H. (2016). Loop closure detection for visual slam using pcanet features. In: *IEEE Joint Conference on Neural Networks* (pp. 2274–2281).

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In: *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1492–1500).

Xin, Z., Cui, X., Zhang, J., Yang, Y., & Wang, Y. (2019). Real-time visual place recognition based on analyzing distribution of multi-scale cnn landmarks. *Journal of Intelligent & Robotic Systems*, *94*, 777–792.

Yu, J., Zhu, C., Zhang, J., Huang, Q., & Tao, D. (2020). Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. *IEEE Transactions on Neural Networks and Learning Systems*, *31*, 661–674.

Zhang, H. (2011). BoRF: Loop-closure detection with scale invariant visual features. In: *IEEE International Conference on Robotics and Automation* (pp. 3125–3130).

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Mueller, J., & Manmatha, R. (2020). Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. *Advances Neural Information Processing Systems*, 487–495.

How to cite this article: An, S., Zhu, H., Wei, D., Tsintotas, K. A., & Gasteratos, A. (2022). Fast and incremental loop closure detection with deep features and proximity graphs. *Journal of Field Robotics*, 1–21. https://doi.org/10.1002/rob.22060