sensors

MDPI

Article

Condition-Invariant Robot Localization Using Global Sequence Alignment of Deep Features

Junghyun Oh [1], Changwan Han [1] and Seunghwan Lee [2,*]

[1] Department of Robotics, Kwangwoon University, Seoul 01897, Korea; jhyunoh@kw.ac.kr (J.O.); hcw511@naver.com (C.H.)
[2] Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Gyeongbuk 39177, Korea
* Correspondence: leesh@kumoh.ac.kr

Abstract: Localization is one of the essential process in robotics, as it plays an important role in autonomous navigation, simultaneous localization, and mapping for mobile robots. As robots perform large-scale and long-term operations, identifying the same locations in a changing environment has become an important problem. In this paper, we describe a robust visual localization system under severe appearance changes. First, a robust feature extraction method based on a deep variational autoencoder is described to calculate the similarity between images. Then, a global sequence alignment is proposed to find the actual trajectory of the robot. To align sequences, local fragments are detected from the similarity matrix and connected using a rectangle chaining algorithm considering the robot's motion constraint. Since the chained fragments provide reliable clues to find the global path, false matches on featureless structures or partial failures during the alignment could be recovered and perform accurate robot localization in changing environments. The presented experimental results demonstrated the benefits of the proposed method, which outperformed existing algorithms in long-term conditions.

Keywords: robotics; localization; sequence alignment; place recognition; deep learning

Citation: Oh, J.; Han, C.; Lee, S. Condition-Invariant Robot Localization Using Global Sequence Alignment of Deep Features. *Sensors* **2021**, *21*, 4103. https://doi.org/10.3390/s21124103

Academic Editors: Guenther Retscher, Ondrej Krejcar, Vassilis Gikas and Michal Kačmařík

Received: 12 May 2021
Accepted: 10 June 2021
Published: 15 June 2021

1. Introduction

Visual place recognition that identifies the same locations between a query and database image sequence is a prerequisite for various robotic applications such as navigation and simultaneous localization and mapping (SLAM) [1–5]. Recent studies have focused on place recognition in changing environments, as autonomous robots should perform large-scale and long-term operations. One of the major challenges for the vision-based place recognition is *appearance changes* caused by variations in weather conditions, time of day, or seasons.

To overcome the appearance change problem, visual place recognition systems can usually be divided into two stages [2,6]. The first stage is a visual front-end that extracts features from the image data to compute similarities between observations, and the second stage is a stochastic back-end that determines the most likely path sequence of a robot by comparing the incoming front-end data. This paper presents a robust feature extraction method using a deep architecture in first part, and a novel sequence alignment algorithm to perform precise localization in second part.

Global descriptors that describe the whole image have been used since they have shown more robust performances than local features in changing conditions [7,8]. Recently, learning-based approaches were widely applied to place recognition under substantial appearance changes, and various deep learning frameworks have been employed to extract features from images [9–13]. In this paper, we present a feature extraction method using a *variational autoencoder* (VAE) [14], one of the powerful deep generative models for feature extraction. Since this model learns to compress the input image in a probabilistic way, the

extracted codes contain useful information and can be used to build a similarity matrix that represents similarity between images.

After generating the similarity matrix, the most likely path sequence of the robot should be estimated from the matrix to perform precise localization. These sequence-based approaches achieved significant improvements in place recognition by attempting to match sequences rather than single images [15,16]. However, they have a common limitation in that if an incorrect local match occurs, the correct global alignment cannot be recovered. To overcome the problem, we propose *glocal sequence alignment*, a combination of the global and local alignment methods, which arranges the sequences of features to perform precise localization.

The overall procedure of the proposed algorithm is shown in Figure 1. Given a similarity matrix by comparing the deep learning features, local fragments that have local maximum scores are detected from the matrix. Then, reliable ones are found by the rectangle chaining algorithm under the motion constraint of the mobile robot. Finally, the most likely path of the robot is determined using the global aligner. As the chained fragments provide reliable clues to find the global path, false matching on featureless structures or partial failures during the alignment could be recovered.

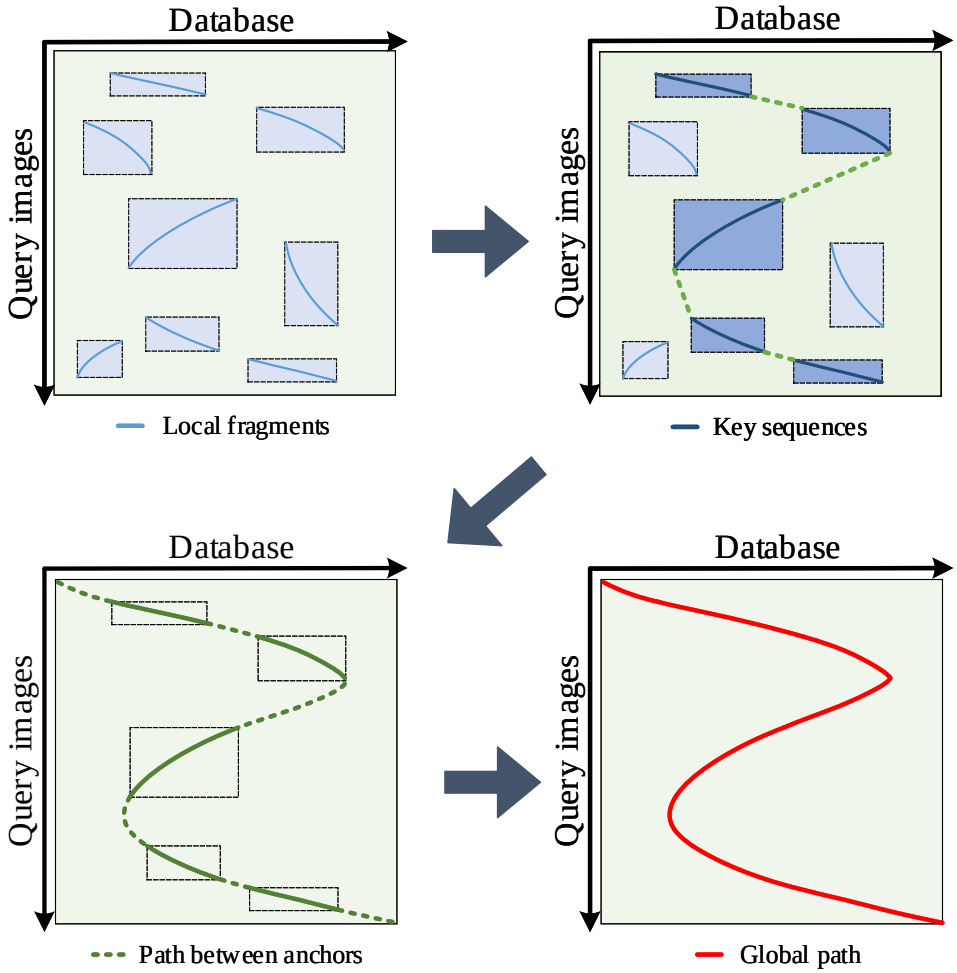

Figure 1. The procedure of the proposed approach. First, local fragments are detected from the similarity matrix using the local alignment method (Section 3.2). Second, the subset of the local alignments is found by the rectangle chaining algorithm that maximizes the total similarity under the motion constraint of the mobile robot (Section 3.3). Then, the path between anchors are calculated using the local alignment method. Finally, the most likely path of the robot is determined using the global aligner (Section 3.4).

The present research paper is organized as follows. Section 2 explains related work on place recognition in changing environments. The proposed methodology is explained in Section 3. The feature extraction method from the VAE and the proposed glocal sequence algorithm is discussed in this section. Section 4 presents the validation of the proposed method through publicly available datasets with other algorithms. Finally, Section 5 concludes the paper.

2. Related Work

The bag-of-words model from local features such as SIFT [17], SURF [18], and BRIEF [19] has been widely applied to visual place recognition tasks [20–22], as they are robust to viewpoint changes. Each image is quantized into a finite number of visual words and can be represented by histograms that can be compared efficiently using Hamming distance or histogram comparison methods. These methods have the advantage of being able to quickly and efficiently recognize a place in a static environment, but have a fatal weakness that false positives can occur in a changing environment.

To overcome the false positive problems, place recognition systems based on global descriptors have been proposed. Unlike the local features, the global descriptors use predefined keypoints and extract information from the whole image. This characteristic makes it possible to distinguish places even if some of image features are similar, and global descriptor based place recognition system has the advantage of being more robust against false positives than local features. Badino et al. [8] proposed whole-image descriptors based on SURF (WI-SURF) to perform place recognition. Similarly, BRIEF-Gist [7] used BRIEF to extract features from the whole image. GIST is one of the popular global descriptors [23] and is widely used in place recognition [24–26]. GIST is based on Gabor filters at different orientations and frequencies to extract various information from the image. These results are averaged to generate a compact meaningful vector. GIST is applied in [24] to capture the basic structure of different types of scenes in a compact way from the portions of panoramic images.

Since deep learning, especially convolutional neural network, showed high performance in image classification and recognition, global image descriptors using CNN have been proposed for visual place recognition [10,27,28]. Naseer et al. extracted a sequence of image descriptors using CNNs to compute the similarity matrix and compute matching hypotheses to find loop closures [27]. Sünderhauf et al. first extracted landmark proposals and utilized CNNs features as landmark descriptors [28]. Performances of the deep learned features are evaluated in [10], and the output features of each layer are compared to find the adequate layer for place recognition. Since those approaches used pretrained CNNs such as AlexNet [29] or LeNet [30] for feature extraction, they showed improved performances in changing environments without requiring any training procedure.

To more actively cope with the changing environment, there have been learning-based methods that directly learn a relationship between environments rather than using the pretrained neural network [9,31]. Neubert et al. proposed the concept of appearance change prediction between two different seasons using vocabularies of superpixels [31]. Despite its novelty, the proposed method relied on handcrafted features and segmentation parameters. Lowry et al. also proposed a supervised and an unsupervised learning method for place recognition in changing environments [9]. The supervised learning method depended on linear regression, which finds a linear transform between the two image sequences to predict environmental changes, and unsupervised learning method tried to remove appearance changes based on principal component analysis (PCA). NetVLAD [32] achieved state-of-the-art performance in place recognition by using the CNN and vector of locally aggregated descriptors (VLAD) but takes a large amount of time to perform model training. Oh and Lee proposed a simple convolutional autoencoder (CAE) to recognize places under extreme perceptual changes [33]. Similar to this idea, this paper proposes a feature extraction method based on a VAE, which is a likelihood-based generative model. Since this structure learns mapping from input data to low-dimensional latent vectors in a

probabilistic way, features from VAE contain a lot of information of the images even in the low-dimensional vector.

After extracting condition-robust features, the most likely path should be determined by finding correspondences between them. Sequence-based approaches are widely used techniques exploiting the temporal information of image sequences [1]. Milford demonstrated that matching a sequence of images rather than a single image achieved improved performances under extreme perceptual changes [16]. However, a critical limitation of the system is a constant velocity assumption which is often violated in practice. To consider speed variations, Naseer et al. proposed minimum cost network flow in a data association graph [34]. Similarly, Viterbi algorithm [35] and dynamic programming (DP) approach [33] were proposed to determine the most likely path through the environment. Recently, DeepSeqSLAM, a trainable architecture combining CNN and a recurrent neural network (RNN), was proposed to learn visual and positional information from an image sequence [36], and SeqNet proposed hierarchical recognition system using learned short sequential descriptors [37]. However, these methods still have two limitations in common: (1) they are likely to find false matches on featureless structures such as tunnels, corridors, and walls and (2) there is no chance to recover the global alignment once incorrect local matches have occurred.

To overcome the problems, a novel *glocal* sequence alignment method for place recognition is presented, inspired by gene sequence matching of bioinformatics [38]. It is a combination of the global and local sequence alignment that can overcome partial failures. As the proposed method first detects reliable parts and calculates the global path by chaining them, it is able to not only find the accurate matches on featureless environments but also recover the global path even if incorrect local matching occurred.

3. Proposed Approach

3.1. Similarity Matrix Generation from Deep Learning Features

VAE is a specific type of a neural network that can compress data into the latent vectors in an unsupervised way. Using the latent vector as an image descriptor, the similarity between images can be calculated. The VAE consists of a standard autoencoder component that embeds the input data $\mathbf{x}$ into latent codes$\mathbf{z}$ by minimizing reconstruction error, and a Bayesian regularization over the latent space, which enforces the posterior of the hidden code vector, matches a prior distribution.

Let us consider the feature $\mathbf{z}$ which compresses the information of the image $\mathbf{I}$. Then the feature $\mathbf{z}$ is assumed to generated from prior distribution $p_\theta(\mathbf{z})$, and an image $\mathbf{I}$ is generated from some conditional distribution $p_\theta(\mathbf{I}|\mathbf{z})$. A recognition model $q_\phi(\mathbf{z}|\mathbf{I})$ which is an approximation to the intractable true posterior $p_\theta(\mathbf{I}|\mathbf{z})$ is introduced to efficiently approximate posterior inference of the latent variable $\mathbf{z}$ given an observed value $\mathbf{I}$ for a choice of parameters θ. The recognition model $q_\phi(\mathbf{z}|\mathbf{I})$ is also referred as a probabilistic *encoder*, since given a datapoint $\mathbf{I}$, it produces a distribution over the possible values of the code $\mathbf{z}$ from which the datapoint $\mathbf{I}$ could have been generated. In a similar vein, $p_\theta(\mathbf{I}|\mathbf{z})$ is a probabilistic *decoder*, since given a code $\mathbf{z}$, it produces a distribution over the possible corresponding values of $\mathbf{I}$. The structure of the VAE is shown in Figure 2.

The loss function $\mathcal{L}(\theta, \phi; \mathbf{I})$ used to train the VAE is the sum of the reconstruction error and the KL-divergence [14] as in the following:

$$\mathcal{L}(\theta, \phi; \mathbf{I}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{I})}[\log p_\theta(\mathbf{I}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{I})||p_\theta(\mathbf{z})). \quad (1)$$

Training is performed to minimize the loss function $\mathcal{L}(\theta, \phi; \mathbf{I})$, and the parameters of the neural network θ and ϕ can be found from solving the optimization problem.

After finishing the training procedure, an input image $\mathbf{I}$ is transformed to reconstruct the output image, and outputs of the intermediate layers $\mathbf{z}$ can be used as the compressed representation of the image. As these features contain entire information of the images, they

are useful for calculating image similarities. If there are two features $\mathbf{z}_i$ and $\mathbf{z}_j$ from different images, the similarity score S_{ij} between them is calculated using the cosine similarity.

$$S_{ij} = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|} \tag{2}$$

Suppose there are M query and N database images from different environments. Then, a similarity matrix $S \in \mathbb{R}^{M \times N}$ can be constructed from these similarity scores where each element S_{ij} is the similarity between two images i and j.

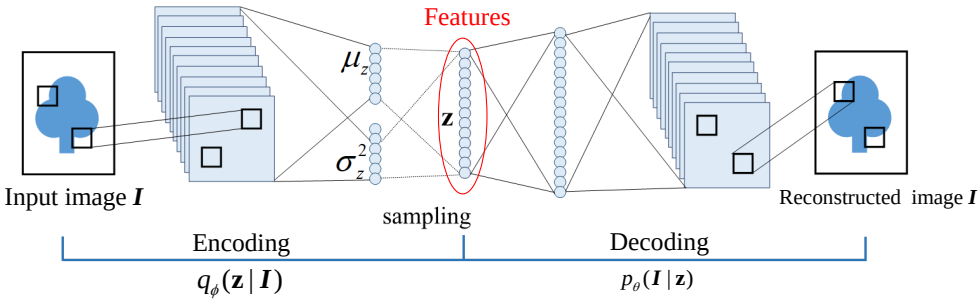

Figure 2. The structure of the VAE that is composed of the encoder and the decoder part.

3.2. Finding Local Fragments from Similarity Matrix

After generating the similarity matrix, local fragments that are candidates for the global path should be detected. To find the fragments, multiple seeds whose similarity scores are above the threshold τ are found from the S. A local sequence alignment method such as Smith–Waterman algorithm [39] is then performed from these points to find local fragments.

The local sequence alignment works through the following procedures. First, the score matrix $\mathbf{H}$ is constructed recursively from the similarity matrix $\mathbf{S}$. $\mathbf{H}$ is initialized with zeros and filled recursively based on the neighbor's similarity score and the gap penalty. In this paper, $\mathbf{H}$ is constructed using the modified version of scoring method in [33]. The main difference is that negative scores are not allocated in our model to enable local alignment as the following:

$$\mathbf{H}_{i,j} = log\mathbf{S}_{i,j} + \max_{k \in W(j)} \left(\mathbf{H}_{i-1,k} + \log \delta(i,j,k), 0 \right) \tag{3}$$

where $W(j) = [j - V_{max}, j + V_{max}]$ is a constrained candidate set, and $\delta(i,j,k)$ is the likelihood of transitioning from state k to state j when the robot's maximum velocity is V_{max}.

Then, the local path is found by tracing back. Starting from the maximum value of the $\mathbf{H}$ to the end of zero, the best local alignment is found by tracking the source of each score recursively. Since we have multiple seeds, multiple local sequences can be detected from the $\mathbf{S}$.

Let the n-th local fragment has the starting point $(x_i(n), y_i(n))$ and the ending point $(x_f(n), y_f(n))$. Then, it can be modeled as a rectangle $R(n)$ with these points as diagonal components and define a weight score $w(n)$ which is a length between these points. A rectangle chain of maximum score should be found given a set of these weighted rectangles.

3.3. Rectangle Chaining Algorithm

Finding a rectangle chain of maximum score is equivalent to find the indices of rectangles $\mathcal{L} = \{p_1, \ldots, p_L\}$, maximizing the global sequence score as the following equation:

$$\mathcal{L} = \arg\max_{p_1, \ldots, p_L} \left(\sum_{l=1}^{L} w(p_l) - \sum_{m=1}^{L-1} \delta(p_m, p_{m+1}) \right) \tag{4}$$

where $\delta(u, v)$ is the gap penalty for connecting $R(u)$ to $R(v)$ in the chain.

To calculate a gap penalty, a new method is proposed considering the motion constraints of the robot. The linear motion model of the robot satisfies $\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{w}$, where $\mathbf{x}$ is the state vector, $\mathbf{F}$ is the state transition matrix, and $\mathbf{w}$ is the process noise drawn from a multivariate Gaussian $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. Let the state vector contains the position and velocity information as $\mathbf{x}_n = \begin{bmatrix}\mathbf{p}_n, \mathbf{v}_n\end{bmatrix}^T$, then the equation can be rewritten as follows:

$$\begin{bmatrix}\mathbf{p}_n \\ \mathbf{v}_n\end{bmatrix} = \begin{bmatrix}\mathbf{I} & t\mathbf{I} \\ \mathbf{0} & \mathbf{I}\end{bmatrix} \begin{bmatrix}\mathbf{p}_{n-1} \\ \mathbf{v}_{n-1}\end{bmatrix} + \mathbf{w} \tag{5}$$

where t is the sampling time. Then, the predicted state after n steps follows $\mathbf{x}_n \sim \mathcal{N}(\mu_n, \mathbf{\Sigma}_n)$ derived as follows:

$$\begin{cases}\boldsymbol{\mu}_n & = \mathbf{F}^n \boldsymbol{\mu}_0 \\ \mathbf{\Sigma}_n & = \mathbf{F}^n \mathbf{\Sigma}_0 (\mathbf{F}^\top)^n + \sum_{k=0}^{n-1} \mathbf{F}^k \mathbf{Q} (\mathbf{F}^\top)^k \end{cases} \tag{6}$$

To calculate the $\delta(u, v)$, the step size n is set to $|y_f(u) - y_i(v)|$, since the initial state of the robot is at the ending point of $R(u)$ and the final state is at the starting point of $R(v)$. Then, the gap penalty finally becomes the following equation:

$$\delta(u, v) = C|\mathbf{\Sigma}_\eta| \exp\Big(-\frac{1}{2}\Big((\mathbf{x}_v - \boldsymbol{\mu}_\eta)^\top \mathbf{\Sigma}_\eta^{-1} (\mathbf{x}_v - \boldsymbol{\mu}_\eta)\Big)^{-1} \tag{7}$$

where $\eta = |y_f(u) - y_i(v)|$, $\mathbf{x}_v$ is the state at $R(v)$, and C is the weight factor. The value of the gap penalty $\delta(u, v)$ becomes larger as the distance between $R(u)$ and $R(v)$ increases. Therefore, the gap should be minimized as possible to maximize the score when chaining the rectangles. The details are described in Appendix A.

The rectangle chaining problem under the gap penalty was first introduced in gene sequence matching [38] to chain the ordered local gene sequences. The idea was based on the *sparse DP* which finds the maximum weight chain by comparing the rectangles in the list. The rectangle chaining algorithm is modified considering the motion constraint of the robot.

A new rectangle is only searched through the query sequence, and the score is evaluated based on the combination of weight scores and the gap penalty. Detail procedure is described in Algorithm 1.

Algorithm 1 Proposed rectangle chaining algorithm.

Input	A set of rectangles $\boldsymbol{R(1), \ldots, R(N)}$
Output	The optimal chaining path $\boldsymbol{P^*}$
1:	**for** $t = 1$ to T
2:	**if** $t = y_i(k)$ of rectangle $R(k)$
3:	$j \leftarrow$ rectangle in $\mathcal{L}$, with largest $y_f(j) < y_i(k)$
4:	$V(k) \leftarrow w(k) + V(j) + \delta(R(j), R(k))$
5:	$P(k) \leftarrow \{k, P(j)\}$
6:	**if** $t = y_f(k)$ of rectangle $R(k)$
7:	$j \leftarrow$ rectangle in $\mathcal{L}$, with largest $y_f(j) \leq y_f(k)$
8:	**if** $V(k) > V(j)$
9:	**Insert** $R(j)$ into $\mathcal{L}$
10:	**Remove** all $R(l)$ with $V(l) \leq V(k)$ and $y_f(l) \geq y_f(k)$
11:	$P^* = P(n^*)$ where $n^* = \arg\max_{n \in [1,N]} V(n)$
12:	**return** P^*

The proposed algorithm has the following improvements compared with the conventional method. First, the searching region for the next rectangle is expanded to deal with both the forward and backward moving of the robot as considered in recent papers [40,41]. Second, the searching direction is changed from the x-axis to y-axis, as the query images comes in time series and the next rectangle should be strictly below the current rectangle.

Finally, the gap penalty is added to consider motion constraint of the robot. The final indices of rectangles $\mathcal{L}$ obtained through this algorithm is a set of *key fragments* that form the global path. Therefore, we can remove other local fragments that can cause a catastrophic failure when finding the global path as shown in Figure 3.

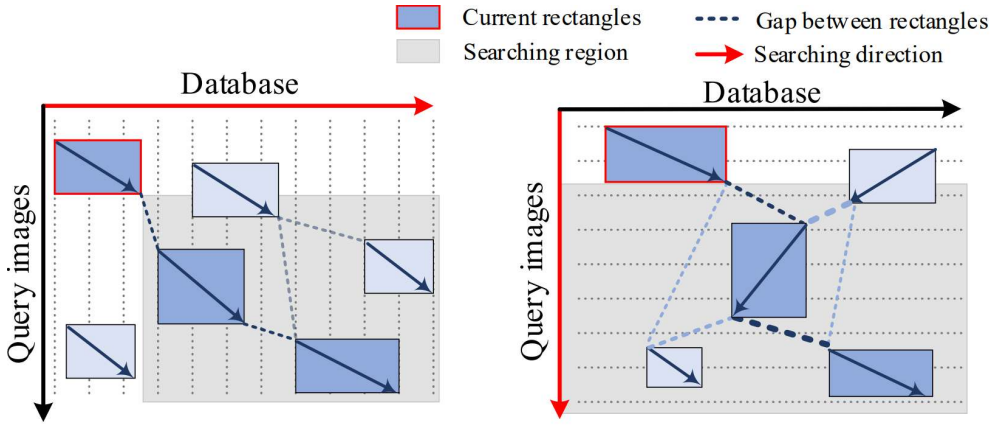

Figure 3. Comparison of existing and proposed rectangle chaining algorithms. The existing sparse DP algorithm (**left**) and the proposed algorithm (**right**). The proposed algorithm not only detects the robot's forward motion but also detects backwardness using a rectangle chaining algorithm.

3.4. Global Sequence Alignment Using Anchors

Finally, the global path should be found by connecting key fragments in $\mathcal{L}$ found in the previous process. The starting and ending point of the key fragments are named *anchors* since they provide reliable clues to find the global path.

Let the set of key fragments is $\mathcal{L} = \{R(p_1), R(p_2), \ldots, R(p_L)\}$. Then, $(L-1)$ paths starting from $(x_f(p_k), y_f(p_k))$ to $(x_i(p_{k+1}), y_f(p_{k+1}))$ should be found, where $k = 1, 2, \ldots, L-1$. It is a kind of global alignment problem because the starting point and the ending point are fixed, and the task is to find a path connecting them. Any existing global alignment method can be used to fill the gap between key fragments in $\mathcal{L}$. Note that we can find more accurate correspondences in the gap because the anchors provide the information of the starting and ending points of the sequences. In this paper, the DP-based method [33], which could consider various robot moving directions, is conducted to find the path between anchors.

Finally, the final global path is determined by the union of paths within the rectangles and paths connecting the anchors. Let the set of paths within rectangles is $\{A_1, A_2, \ldots, A_L\}$ and the set of paths connecting the anchors is $\{B_1, B_2, \ldots, B_{L-1}\}$. Then, the final global path is determined by alternatively concatenating the local sequences in key fragments and aligned sequences connecting the anchors as $\{A_1, B_1, \ldots, A_{L-1}, B_{L-1}, A_L\}$. The proposed method has the advantage in overcoming false matches on featureless structures or partial failures during the alignment because it detects the paths with high reliability first and then connects them to find the final path. Therefore, it can perform accurate robot localization in changing environments.

4. Experimental Evaluation

4.1. Experimental Setup

To demonstrate the effectiveness of the proposed approach, three datasets were used from diverse environments. Since the proposed method aims to verify the performance of place recognition in changing conditions, the viewpoint changing problem is not considered. The Alderley dataset [15] consists of data collected during the days and nights on the same route. The daytime traverse was used as the database images and one nighttime traverse was used as the query image. The Oxford RobotCar dataset [42] consists of images taken on a sunny day and a rainy night in the city. The central images from the traverse ids 2014-11-25-09-18-32 and 2014-11-21-16-07-03 were used as database and query images, respectively. The final dataset is the Nordland dataset [43] collected from four seasons of

rail journey. The spring–winter pair was used for our experiment. All images were resized to 224 × 224 and aligned to the same locations. In each dataset, 6000 images were used as a training set and 500 images were used as a test set. The sample image sequence of the datasets are shown in Figure 4. All experiments were implemented in Python 3.6.9 using the libraries Tensorflow 2.5.0 and carried out on a PC with a 3.9 GHz Intel Core-i7 CPU and 16 GB of main memory.

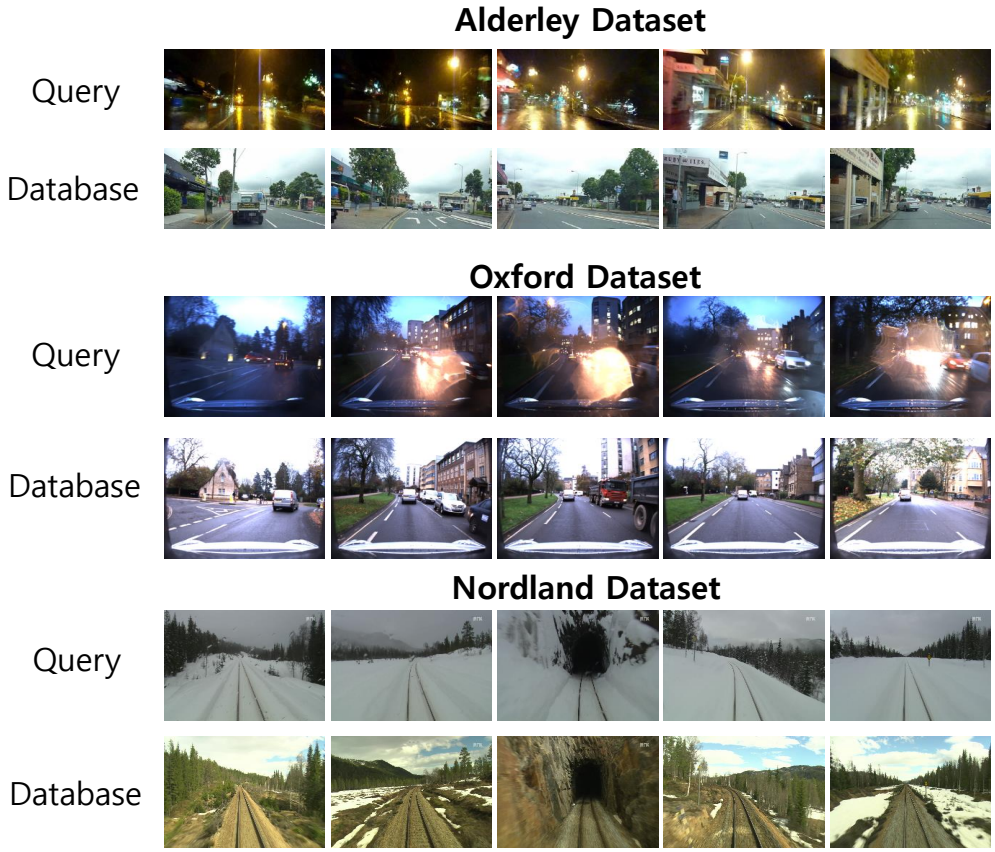

Figure 4. Sample image sequence taken from each of the 3 datasets: Alderley (day-to-night), Oxford (sunny-to-rainy), and Nordland (spring-to-winter).

Our contributions are to propose a robust feature extraction based on VAE and a global alignment algorithm using the extracted features, so we divided the experiment into two main parts, the precision–recall performance of the features and the global alignment performances.

4.2. The Precision–Recall Performance of the Features

In the first part, the place recognition performance of the proposed feature was verified through precision–recall analysis. The relationship between precision and recall can be represented by a precision–recall curve, and the area under the curve (AUC) is a widely used metric for evaluating the place recognition performances. Precision is defined as the percentage of the number of correct matches for the number of total matches detected, and recall is the ratio of the number of correct matches to the total number of matches, that is:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{8}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{9}$$

Our feature was compared with sum-of-absolute differences (SAD) from SeqSLAM [15], AlexNet [29], and NetVLAD [32]. The compared features demonstrated robust performance in changing environments, and widely used features in place recognition. Our proposed features were extracted from VAE model as shown in Table 1. Since the place recognition performance depends on the number of layers and nodes in the network, the best parameters were chosen through experiments.

Table 1. The output shape of each layer in our VAE model.

Layer	Size	Layer	Size	Layer	Size	Layer	Size
`conv1`	$112 \times 112 \times 32$	`conv4`	$14 \times 14 \times 128$	`fc7`	2048	`z_mean`	128
`conv2`	$56 \times 56 \times 64$	`conv5`	$7 \times 7 \times 128$	`fc8`	1024	`z_var`	128
`conv3`	$28 \times 28 \times 64$	`fc6`	4096	`fc9`	512	`sampling`	128

The precision–recall results for each dataset are shown in Figure 5. In our test set, there is little difference in distance between frames, and as the environment changes extremely, most of the features do not exhibit high performance. However, our proposed feature showed comparable performance even when compared to the state-of-the-art feature, NetVLAD, and showed superior performance in the Oxford dataset. Since all feature extraction algorithms used the pretrained network, there was no significant difference in processing time.

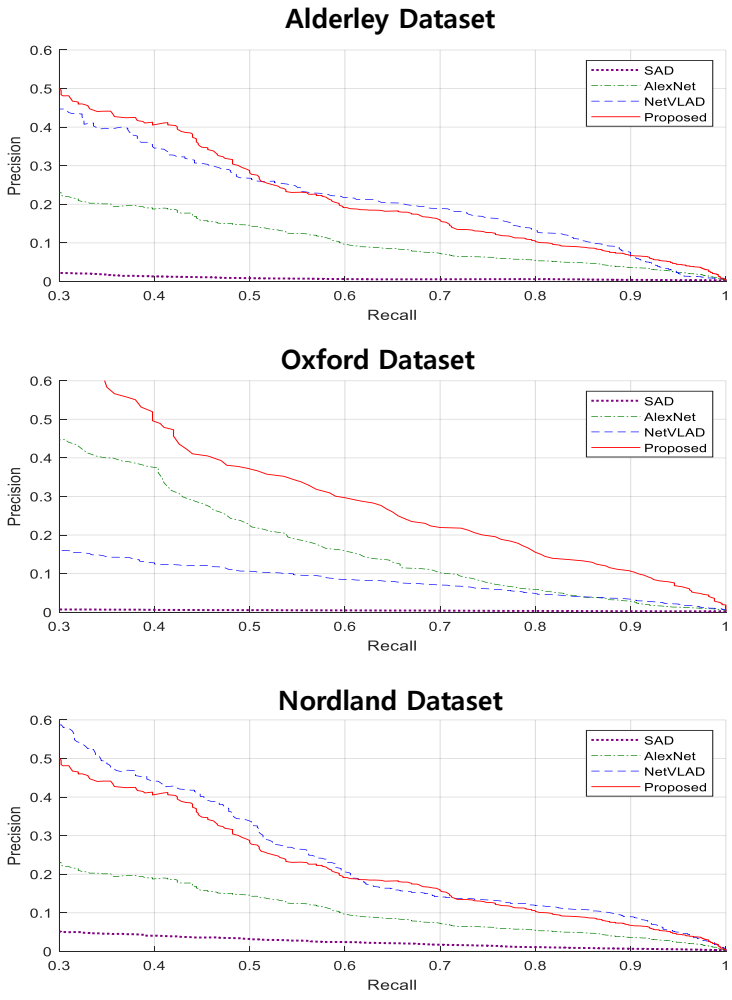

Figure 5. Sample image sequence taken from each of the three datasets: Alderley (day-to-night), Oxford (sunny-to-rainy), and Nordland (spring-to-winter).

The reason why our proposed feature performed so well on the Oxford dataset is that the images in this dataset were gray images, so the dimension was low. Our feature's dimension is 128, as shown in Table 2, which is a very small number to contain the information of the whole image. However, compared to other features, it can be seen that our feature can efficiently store a lot of information despite its low dimensionality. Therefore, increasing the number of layers or nodes in the VAE can achieve high performance even with images of large dimensions.

In this experiment, the deep features from VAE were used, but there were limitations. Therefore, it is necessary to improve performance in combination with sequence-based global alignment method using the robot's movement information.

4.3. Global Alignment Performance

To evaluate the global alignment performance, experiments were conducted on the Nordland dataset. We used the spring and winter sequences, and they were rearranged to generate various situations such as acceleration, deceleration, reverse moving, etc. The ground truth of the corresponding frames and examples of matched frames were shown in Figure 6.

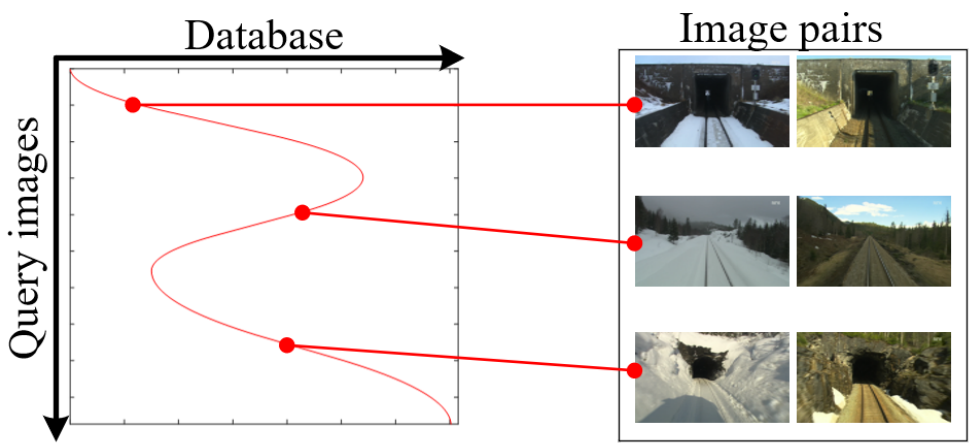

Figure 6. Ground truth of the corresponding frames and examples of matched frames.

Experimental results are shown in Figure 7. First, local seeds were detected above the similarity 0.99 and performed local sequence alignment using the DP [33]. The local fragments above the weight scores 50 are chosen to be the candidates for the global alignment as shown in Figure 7b. Then, they are modeled as rectangles and connected using the proposed rectangle chaining algorithm. In Figure 7c, key sequences are represented as blue lines and the connections are shown as green dotted lines. Other local fragments represented as black lines are unnecessary. Finally, the global path estimated using the key sequences as anchors is shown in Figure 7d. To compare the performances of the proposed method, the resulting path of the SeqSLAM [15] and DP [33] are also presented.

We can conclude that the proposed method outperforms other algorithms, as SeqSLAM showed an inaccurate path due to the assumption of constant speed, and the DP partially failed to estimate the path in repetitive structures such as tunnels and roads. The precision–recall results and F1-scores in Table 2 also showed that the proposed method is more accurate than other methods.

Table 2. The precision–recall and F1-score results.

Method	SeqSLAM [15]	DP [33]	Proposed
Precision	0.010	0.296	0.955
Recall	0.012	0.288	0.957
F1-score	0.011	0.292	0.956

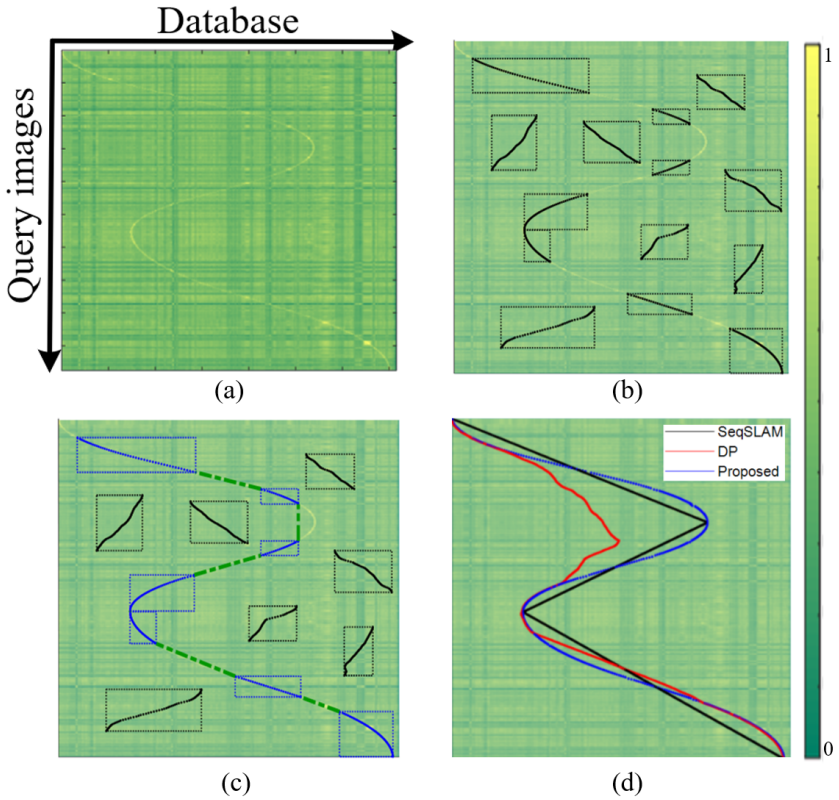

Figure 7. Experimental results: (**a**) constructed similarity matrix; (**b**) detected local fragments; (**c**) rectangle chained results; (**d**) global alignment results compared to other algorithms.

5. Conclusions

To achieve robot localization in changing environments, the robust feature extraction method using the variational autoencoder was described to calculate the similarities between images. Then, the global sequence alignment method based on sparse DP was proposed to chain the reliable local fragments under the motion constraint of the robot. Experiments were performed on three datasets to demonstrate the effectiveness of the proposed approach. First, a precision–recall analysis was performed to test the robustness of the deep features, and the experimental results showed that the proposed features showed stable performance in various environments. In the Oxford dataset, the F1-score—which is the harmonic mean of the precision and recall—achieved 30% higher than that of AlexNet. In other datasets, the proposed feature achieved precision–recall results comparable to NetVLAD. Second, the global alignment performances were tested on the rearranged Nordland dataset. The false matches during the alignment were recovered, and the path of the robot was successfully estimated by using the proposed method. The precision–recall results showed that our method achieved more than three times higher performance than other methods.

Although the proposed method showed improved place recognition performance in appearance changing environments, another challenging environment in robot localization is the *viewpoint changing environment*. The viewpoint of the same place can change drastically when revisiting it, and finding correspondences between database and query images in this situation is challenging. Since the proposed autoencoder feature is a kind of global descriptors, it has limitations in dealing with viewpoint change problems compared to other local descriptor-based methods. In the future, it is necessary to improve our method to overcome the appearance changing problem as well as the viewpoint changing problem for practical robot localization.

Author Contributions: Conceptualization, J.O.; methodology, J.O.; validation, J.O. and C.H.; formal analysis, J.O.; investigation, J.O. and S.L.; project administration, J.O.; funding acquisition, J.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work has supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No. 2020R1F1A1076667), Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20174010201620). Also, this work was supported by Research Resettlement Fund for the new faculty of Kwangwoon University in 2019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SLAM	simultaneous localization and mapping
VAE	variational auto encoder
CAE	convolutional auto encoder
VLAD	vector of locally aggregated descriptors
CNN	convolutional neural network
RNN	recurrent neural network
DP	dynamic programming
SAD	sum of absolute differences

Appendix A. Derivation of the Gap Penalty

The robot motion model assumes the true state at step n is evolved from the step at $n-1$ according to the following equation:

$$\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{B}\mathbf{u}_n + \mathbf{w} \tag{A1}$$

where $\mathbf{F}$ is the state transition matrix, $\mathbf{B}$ is the control model, $\mathbf{w}$ is the process noise which is assumed to be drawn from a multivariate normal distribution $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$.

In the model, the state vector is defined to have a position and velocity information as $\mathbf{x}_n = [\mathbf{p}_n, \mathbf{v}_n]$. Then, the linear motion model satisfies the following equation:

$$\begin{bmatrix} \mathbf{p}_n \\ \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{I} & t\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{n-1} \\ \mathbf{v}_{n-1} \end{bmatrix} + \mathbf{w} \tag{A2}$$

where t is the sampling time. Then, n-steps after predicted states and covariances are as follows:

$$\begin{aligned} \boldsymbol{\mu}_n &= \mathbf{F}\boldsymbol{\mu}_{n-1} \\ &= \mathbf{F}^n\boldsymbol{\mu}_0 \\ \boldsymbol{\Sigma}_n &= \mathbf{F}\boldsymbol{\Sigma}_{n-1}\mathbf{F}^\top + \mathbf{Q} \\ &= \mathbf{F}^n\boldsymbol{\Sigma}_0(\mathbf{F}^\top)^n + \sum_{k=0}^{n-1} \mathbf{F}^k\mathbf{Q}(\mathbf{F}^\top)^k. \end{aligned} \tag{A3}$$

As the predicted state after n steps follows $\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, the probability of the current state when given initial state and the number of steps after initial state is calculated.

Our purpose is to calculate the gap penalty between rectangles as shown in Figure A1. From the view of the $R(a)$, the purpose is to calculate the gap penalty between $R(b)$, $R(c)$,

and $R(d)$. As $R(a)$ is the initial state, the expected position of the robot can be seen by the evolution of the Gaussian distribution.

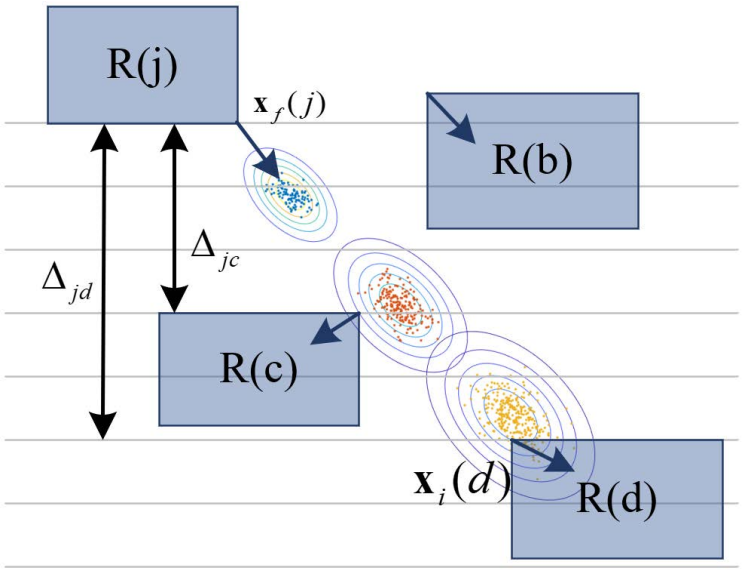

Figure A1. Gap penalty calculation using the linear robot motion model.

The initial state is at the ending point of $R(j)$, which is $\mathbf{x}_0 = [\mathbf{p}_f(j), \mathbf{v}_f(j)]^T$ and the final state is at the starting point of the candidate rectangle $R(k)$, $\mathbf{x} = [\mathbf{p}_i(k), \mathbf{v}_i(k)]^T$. Therefore, the step size is $\Delta = |y_f(j) - y_i(k)|$, the gap penalty becomes the following equation:

$$\delta(R(j), R(k)) = C|\Sigma_\Delta| \exp\left(-\frac{1}{2}\left((\mathbf{x} - \boldsymbol{\mu}_\Delta)^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_\Delta)\right)^{-1} \right. \tag{A4}$$

where $\mathbf{x}$ is the state at $R(k)$ and C is the weight factor.

Let us consider the Figure A1 case. $R(b)$ does not need to be considered since $y_f(j) < y_i(d)$. In the case of $R(c)$, the Gaussian distribution corresponding to the Δ_{jc} is represented as red dots. However, the state $\mathbf{x}_c = [\mathbf{p}_i(c), \mathbf{v}_i(c)]^T$ is far from the center of the Gaussian distribution, and it can be predicted that it will have a low probability. On the other hand, the Gaussian distribution corresponding to the Δ_{jd} is represented as yellow dots in the case of $R(d)$, and it can be found that the state $\mathbf{x}_d = [\mathbf{p}_i(d), \mathbf{v}_i(d)]^T$ is near the center of the distribution. The penalty is defined as the inverse of the probability, and it is concluded that $\delta(R(j), R(c)) > \delta(R(j), R(d))$.

References

1. Lowry, S.; Sünderhauf, N.; Newman, P.; Leonard, J.J.; Cox, D.; Corke, P.; Milford, M.J. Visual place recognition: A survey. *IEEE Trans. Robot.* **2016**, *32*, 1–19. [CrossRef]
2. Zeng, Z.; Zhang, J.; Wang, X.; Chen, Y.; Zhu, C. Place recognition: An overview of vision perspective. *Appl. Sci.* **2018**, *8*, 2257. [CrossRef]
3. López, E.; García, S.; Barea, R.; Bergasa, L.M.; Molinos, E.J.; Arroyo, R.; Romera, E.; Pardo, S. A multi-sensorial simultaneous localization and mapping (SLAM) system for low-cost micro aerial vehicles in GPS-denied environments. *Sensors* **2017**, *17*, 802. [CrossRef]
4. Marchel, Ł.; Naus, K.; Specht, M. Optimisation of the Position of Navigational Aids for the Purposes of SLAM technology for Accuracy of Vessel Positioning. *J. Navig.* **2020**, *73*, 282–295. [CrossRef]
5. Yuan, X.; Martínez-Ortega, J.F.; Fernández, J.A.S.; Eckert, M. AEKF-SLAM: A new algorithm for robotic underwater navigation. *Sensors* **2017**, *17*, 1174. [CrossRef] [PubMed]
6. Lowry, S.M. Visual Place Recognition for Persistent Robot Navigation in Changing Environments. Ph.D. Thesis, Queensland University of Technology, Brisbane, Australia, 2014.
7. Sünderhauf, N.; Protzel, P. BRIEF-Gist—Closing the loop by simple means. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Brisbane, Australia, 25–30 September 2011; pp. 1234–1241. [CrossRef]

8. Badino, H.; Huber, D.; Kanade, T. Real-time topometric localization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 1635–1642. [CrossRef]
9. Lowry, S.; Milford, M. Supervised and unsupervised linear learning techniques for visual place recognition in changing environments. *IEEE Trans. Robot.* **2016**, *32*, 600–613. [CrossRef]
10. Sünderhauf, N.; Shirazi, S.; Dayoub, F.; Upcroft, B.; Milford, M. On the performance of ConvNet features for place recognition. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 4297–4304.
11. Chen, Z.; Jacobson, A.; Sünderhauf, N.; Upcroft, B.; Liu, L.; Shen, C.; Reid, I.; Milford, M. Deep learning features at scale for visual place recognition. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3223–3230.
12. Latif, Y.; Garg, R.; Milford, M.; Reid, I. Addressing Challenging Place Recognition Tasks Using Generative Adversarial Networks. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2349–2355.
13. Mao, J.; Hu, X.; He, X.; Zhang, L.; Wu, L.; Milford, M.J. Learning to Fuse Multiscale Features for Visual Place Recognition. *IEEE Access* **2019**, *7*, 5723–5735. [CrossRef]
14. Kingma, D.; Welling, M. Auto-encoding variational Bayes. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
15. Milford, M.; Wyeth, G. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 1643–1649.
16. Milford, M. Vision-based place recognition: How low can you go? *Int. J. Robot. Res.* **2013**, *32*, 766–789. [CrossRef]
17. Lowe, D. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
18. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
19. Calonder, M.; Lepetit, V.; Ozuysal, M.; Trzcinski, T.; Strecha, C.; Fua, P. BRIEF: Computing a local binary descriptor very fast. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1281–1298. [CrossRef] [PubMed]
20. Angeli, A.; Filliat, D.; Doncieux, S.; Meyer, J.A. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Trans. Robot.* **2008**, *24*, 1027–1037. [CrossRef]
21. Cummins, M.; Newman, P. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Int. J. Robot. Res.* **2008**, *27*, 647–665. [CrossRef]
22. Cummins, M.; Newman, P. Appearance-only SLAM at large scale with FAB-MAP 2.0. *Int. J. Robot. Res.* **2011**, *30*, 1100–1123. [CrossRef]
23. Oliva, A.; Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
24. Murillo, A.; Košecká, J. Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons. In Proceedings of the Presented at the IEEE International Conference on Computer Vision (ICCV) Workshops, Kyoto, Japan, 27 September–4 October 2009.
25. Siagian, C.; Itti, L. Biologically inspired mobile robot vision localization. *IEEE Trans. Robot.* **2009**, *25*, 861–873. [CrossRef]
26. Liu, Y.; Zhang, H. Visual loop closure detection with a compact image descriptor. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Algarve, 7–12 October 2012; pp. 1051–1056.
27. Naseer, T.; Ruhnke, M.; Stachniss, C.; Spinello, L.; Burgard, W. Robust visual SLAM across seasons. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 2529–2535.
28. Sünderhauf, N.; Shirazi, S.; Jacobson, A.; Dayoub, F.; Pepperell, E.; Upcroft, B.; Milford, M. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In Proceedings of the Robotics: Science and Systems XI:. Robotics: Science and Systems Conference, Rome, Italy, 13–17 July 2015; pp. 1–10.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. In Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, 3–6 December 2012; pp. 1097–1105.
30. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
31. Neubert, P.; Sünderhauf, N.; Protzel, P. Appearance change prediction for long-term navigation across seasons. In Proceedings of the European Conference on Mobile Robots, Barcelona, Spain, 25–27 September 2013; pp. 198–203.
32. Arandjelović, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
33. Oh, J.H.; Lee, B.H. Dynamic programming approach to visual place recognition in changing environments. *Electron. Lett.* **2017**, *53*, 391–393. [CrossRef]
34. Naseer, T.; Spinello, L.; Burgard, W.; Stachniss, C. Robust visual robot localization across seasons using network flows. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada 27–31 July 2014; pp. 2564–2570.

35. Hansen, P.; Browning, B. Visual place recognition using HMM sequence matching. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, USA, 14–18 September 2014; pp. 4549–4555.
36. Chancán, M.; Milford, M. DeepSeqSLAM: A Trainable CNN+RNN for Joint Global Description and Sequence-based Place Recognition. *arXiv* **2020**, arXiv:2011.08518.
37. Garg, S.; Milford, M. SeqNet: Learning Descriptors for Sequence-Based Hierarchical Place Recognition. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4305–4312. [CrossRef]
38. Brudno, M.; Malde, S.; Poliakov, A.; Do, C.B.; Couronne, O.; Dubchak, I.; Batzoglou, S. Glocal alignment: Finding rearrangements during alignment. *Bioinformatics* **2003**, *19*, i54–i62. [CrossRef]
39. Smith, T.F.; Waterman, M.S. Identification of common molecular subsequences. *J. Mol. Biol.* **1981**, *147*, 195–197. [CrossRef]
40. Garg, S.; Suenderhauf, N.; Milford, M. Don't look back: Robustifying place categorization for viewpoint-and condition-invariant place recognition. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3645–3652.
41. Gadd, M.; De Martini, D.; Newman, P. Look around you: Sequence-based radar place recognition with learned rotational invariance. In Proceedings of the 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, ON, USA, 20–23 April 2020; pp. 270–276.
42. Maddern, W.; Pascoe, G.; Linegar, C.; Newman, P. 1 year, 1000 km: The Oxford RobotCar dataset. *Int. J. Robot. Res.* **2017**, *36*, 3–15. [CrossRef]
43. Sünderhauf, N.; Neubert, P.; Protzel, P. Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons. In Proceedings of the Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6 May 2013.