

COMPUTER IMPLEMENTED METHOD FOR DETERMINING SEQUENCE TRAJECTORY AND VISUAL PLACE RECOGNITION METHOD

Field of the invention

- 5 The present invention relates to a computer implemented method for determining a sequence trajectory of a scene for use in visual place recognition. The present invention further relates to a method for visual place recognition and related systems and methods.

Background

- 10 Visual place recognition (VPR) is a core technique of many promising applications in the field of computer vision and robotics, such as autonomous driving, geolocalization, 3D reconstruction, and virtual reality. VPR may be described as a task to determine if a location has been visited and saved in a database before.

- However, this task has been difficult, and many obstacles remain unresolved in this field. For instance, scenes in a city-scale environment are composed of similar entities (e.g., buildings, roads, vegetation) with consistent spatial layouts. Further, the same environment may undergo significant variations in appearance due to changes in illumination, seasons, and weather. Several learning-based methods may deal with this problem by aggregating features that are robust to such significant variations in appearance.
- 15
- 20

- Moreover, the scenes may contain dynamic objects, such as a person or a vehicle, that are likely to move or change frequently in the city-like environment but are not indicative of a particular location. Such dynamic objects may confuse a VPR system. Therefore, some recent methods proposed to leverage attention modules to emphasize on static objects that are stationary and are unlikely to ever move (e.g., buildings, roads, traffic signs, etc.) while reducing the impact of the dynamic objects.
- 25

VPR as a single-frame retrieval (e.g., SeqSLAM) can bring decent performance and efficiency. However, the input source for unmanned vehicles and autonomous robots is usually video or continuous image frames. Decoupling temporal

relationship between input frames may cause loss of information and may weaken the VPR system when dealing with some challenging scenarios. Therefore, sequence-based methods proposed to search image sequences in a database and obtained reliable performance. However, they are mainly based on an ideal hypothesis that the input frames are taken at a constant speed, which may not always be the case in practice. Vehicles or robots often move at non-uniform speeds, especially at intersections or areas with many obstacles.

In some cases, methods may rely on either artificial rule or uniform thresholds to determine if a retrieved trajectory depicts the same place of the query. However, they are not universally applicable to all scenarios. For example, there is a much greater chance of falsely triggering a loop closure on similar looking roads without obvious landmarks.

Therefore, there remains a need for a method that is robust to one or more of the challenges stated above, is adaptive and dynamic, and is universally applicable to all scenarios.

Summary

It is the object of the present invention to improve visual place recognition (VPR) methods.

The object is achieved by the subject-matter of the independent claims. Preferred embodiments are subject-matter of the dependent claims.

The present invention provides a computer implemented method for determining a sequence trajectory of a scene for use in visual place recognition. The method comprises: a) determining a difference matrix having a plurality of difference scores by comparing each image descriptor in a cache sequence with each image descriptor in a database sequence, each image descriptor corresponding to a respective image of the scene; b) determining a plurality of sequence blocks from the difference matrix for obtaining a plurality of sequence trajectories corresponding to the plurality of sequence blocks, each of the plurality of sequence blocks being a sub-set of the difference matrix; c) selecting a difference score having a minimum value in each of the plurality of sequence blocks as a starting point of the

corresponding sequence trajectory; d) selecting a difference score having a minimum value within a predetermined neighbourhood range of a current sequence point in each of the plurality of sequence blocks as a next sequential point of the corresponding sequence trajectory, the current sequence point being the starting point of the corresponding sequence trajectory when the sequential point next to the starting point is determined; e) repeating step d) to determine the sequential points of the corresponding sequence trajectory until the plurality of sequence trajectories are obtained; and f) selecting a final sequence trajectory from the plurality of sequence trajectories.

- 10 Since the method includes selecting the difference score having the minimum value in each of the plurality of sequence blocks as the starting point and selecting the difference score having the minimum value among the neighbours within a certain range to be the next sequential point of the trajectory, a curvilinear trajectory with the sequential points having lowest adjacent difference scores is established
- 15 incrementally. Therefore, the method may adaptively and dynamically alter a step size of adjacent points in the corresponding sequence trajectory. In other words, the method may adaptively alter the stride between the consecutive sequential points.

Further, as the method does not use a predefined uniform velocity to retrieve sequences, vehicles, or robots, that generally move at non-uniform speeds, may

20 therefore find a best matched sequence trajectory from the database sequence using the method of the present invention.

It is preferred, that steps d) to e) are repeated for opposing sides of the starting point to determine the sequential points on the opposing sides of the starting point until the corresponding sequence trajectory has a predefined trajectory length.

- 25 Therefore, the sequential points on the opposing sides of the starting point correspond to both the future and the past frames of the starting point.

It is preferred, that in step b) the difference matrix is sub-sampled by a minimal pooling layer to generate a plurality of minimal blocks, each of the plurality of minimal blocks having a minimum difference score. A predetermined number of minimal

blocks having the least minimal difference score is selected from the plurality of minimal blocks as the plurality of sequence blocks.

5 The selection of the predetermined number of minimal blocks as the plurality of sequence blocks may be based on desired application attributes. This may help to reduce and/or adjust computational complexity of the method based on the desired application attributes.

It is preferred, that the cache sequence is a stack having a predefined stack length, wherein the predefined trajectory length is equal to the predefined stack length.

10 Since the predefined trajectory length is equal to the predefined stack length, the predefined stack length may be adjusted based on a desired trajectory length.

It is preferred, that the minimal pooling layer has a kernel size equal to the predefined stack length.

15 The minimal pooling layer with the kernel size equal to the predefined stack length may prevent adjacent blocks from overlapping and may determine the plurality of minimal blocks faster.

It is preferred, that an image descriptor of an image of the scene is added to the cache sequence, such that the image descriptor of the image is added after an image descriptor of a preceding image of the scene.

20 Therefore, the image descriptors of respective images are sequentially added to the cache sequence.

It is preferred, that the image descriptor is removed from the cache sequence and added to the database sequence when a number of successive image descriptors added to the cache sequence after the image descriptor is equal to the predefined stack length.

25 This may maintain the predefined stack length of the cache sequence.

It is preferred, that the predetermined neighbourhood range is located along a column of the corresponding sequence block adjacent to the current sequence point.

The limited area of the predetermined neighbourhood range may ensure real-time and high efficiency for determining the corresponding sequence trajectory.

It is preferred, that in step f) a sum of the difference scores of the starting point and the sequential points of the corresponding sequence trajectory is determined for
5 each of the plurality of sequence trajectories, wherein a sequence trajectory having a minimum value of the sum of the difference scores is selected from the plurality of sequence trajectories as the final sequence trajectory.

The sequence trajectory having the minimum value of the sum of the difference scores may be a best matched sequence trajectory from the database sequence.
10 Thus, the final sequence trajectory may be the best matched sequence trajectory.

It is preferred, that a Sobel filter is applied to the difference matrix to obtain a difference saliency matrix.

The Sobel filter may be used to obtain the difference saliency matrix in a simple and a time-efficient manner.

15 It is preferred, that a portion of the difference matrix corresponding to the final sequence trajectory and a portion of the difference saliency matrix corresponding to the final sequence trajectory are concatenated to form a concatenated matrix, wherein a convolutional neural network (CNN)-based discriminator is applied to the concatenated matrix to generate a discriminator output.

20 The CNN-based discriminator may learn an optimal judgmental inference from datasets or learned statistics in a data-driven manner in order to determine the confidence of the final sequence trajectory. Thus, the CNN-based discriminator may evaluate confidence of the final sequence trajectory efficiently and accurately.

It is preferred, that a softmax function is applied to the discriminator output to obtain
25 a confidence score of the final sequence trajectory.

The confidence score of the final sequence trajectory may help to validate if the final sequence trajectory is a true correspondence. This may significantly alleviate the false-positive and the false-negative retrievals.

It is preferred, that in step a) each of the plurality of difference scores is equal to a difference between the corresponding image descriptor in the cache sequence and the corresponding image descriptor in the database sequence.

5 A lower difference score of a corresponding image descriptor in the cache sequence and a corresponding image descriptor in the database sequence may indicate a greater similarity between the images of the corresponding image descriptor in the cache sequence and the corresponding image descriptor in the database sequence.

The present invention further provides a method for visual place recognition. The method comprises: 1) obtaining a plurality of images of a scene in an input
10 sequence; 2) performing a preferred method for determining a sequence trajectory of a scene on the plurality of images to obtain a final sequence trajectory of the scene; and 3) generating a control signal that causes a device to perform an action based on the final sequence trajectory.

The method for visual place recognition may therefore find a best matched sequence
15 trajectory from the database sequence and further generate the control signal for the device to perform the action for applications, such as autonomous movement/driving, geolocalization, 3D reconstruction, and virtual reality.

The present invention further provides a vehicle having an image sensor configured to obtain a plurality of images of a scene; a visual place recognition device that is
20 configured for performing a preferred method, so as to obtain a control signal; and a control unit that is configured to process the control signal and perform some action based on the control signal.

The visual place recognition device may therefore find a best matched sequence trajectory from the database sequence and further generate the control signal for the
25 control unit to perform some action, such as steering, braking, and accelerating the vehicle.

The present invention further provides a computer program, a machine-readable storage medium, or a data signal that includes instructions that, upon execution on a data processing device, cause the device to perform one, some, or all of the steps of
30 a method according to the present invention.

The computer program, the machine-readable storage medium, or the data signal may therefore find a best matched sequence trajectory from the database sequence and further generate the control signal for the device to perform the action for applications, such as autonomous movement/driving, geolocalization, 3D

5 reconstruction, and virtual reality.

Brief description of the drawings

Embodiments will now be described by way of example only, with reference to the Figures, in which:

Fig. 1 illustrates a schematic block diagram of an example of a visual place
10 recognition device;

Figs. 2A-2O illustrate schematic diagrams of various exemplary steps performed by an adaptive sequence search module of the visual place recognition device;

Fig. 3 illustrates a schematic diagram of an exemplary step performed by a Sobel filter on an output of the adaptive sequence search module;

15 Figs. 4A-4C illustrate schematic diagrams of exemplary steps to obtain a concatenated matrix;

Fig. 5 illustrates a schematic block diagram of an example architecture of a convolutional neural network (CNN)-based discriminator;

Fig. 6 illustrates an example flow chart for a computer implemented method for
20 determining a sequence trajectory of a scene for use in visual place recognition;

Figs. 7A-7B depict example sequence trajectories; and

Fig. 8 illustrates an example plot depicting various precision-recall curves of datasets using AdaptSeq with and without verification by the CNN-based discriminator.

Detailed description

Aspects and embodiments of the present invention will now be discussed with reference to the accompanying figures. Further aspects and embodiments will be apparent to those skilled in the art.

5 Fig. 1 depicts a schematic block diagram of an example of a visual place recognition device 550. The visual place recognition device 550 includes one or more encoders 10 and an adaptive sequence search module 100. The visual place recognition device 550 may further include a convolutional neural network (CNN)-based discriminator 430.

10 The visual place recognition device 550 is configured to obtain a plurality of images 102 of a scene 50 in an input sequence 105. In the illustrated example of Fig. 1, the visual place recognition device 550 includes a plurality of encoders 10 corresponding to the plurality of images 102.

The visual place recognition device 550 is configured to perform one or more
15 operations to obtain a final sequence trajectory 350 of the scene 50. In some cases, the visual place recognition device 550 is configured to generate a control signal 555 that causes a device 600 to perform an action based on the final sequence trajectory 350. In some cases, the device may be a control unit.

In some cases, a vehicle (not shown) may include an image sensor (not shown)
20 configured to obtain the plurality of images 102 of the scene 50. The image sensor may be mounted on the vehicle. Further, the vehicle may include the visual place recognition device 550 so as to obtain the control signal 555. The vehicle may also include the device 600 or the control unit that is configured to process the control signal 555 and perform the action based on the control signal 555. In some cases,
25 the vehicle may be an unmanned vehicle or an autonomous vehicle. In some cases, an autonomous robot may include the image sensor, the visual place recognition device 550, and the device 600.

Each of the one or more encoders 10 is configured to determine an image descriptor 104 for a respective image 102 from the plurality of images 102 of the scene 50.

Each image descriptor 104 corresponds to the respective image 102 of the scene 50.

Each of the one or more encoders 10 may include a convolutional neural network (not shown) that processes the image 102 for extraction of local features. Further, the local features may be clustered into K visual word clusters through a soft-assignment weight. Additionally, an intra-cluster saliency weight may be applied to further suppress task irrelevant features in each cluster. Further, word vectors may be obtained by aggregating the double weighted (i.e., based on the soft-assignment and the intra-cluster saliency weights) local residuals. Furthermore, the image descriptor 104 may be generated by concatenating normalized visual word vectors and scaling them by an L2-normalized cluster saliency. The L2-normalized cluster saliency may be obtained from a set of trainable weights.

The details of methods for determining the image descriptor 104 are described in WO 2022/258624 A1 and G. Peng, J. Zhang, H. Li and D. Wang, "Attentional Pyramid Pooling of Salient Visual Residuals for Place Recognition," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 865-874, doi:10.1109/ICCV48922.2021.00092, the contents of the disclosures being incorporated by reference in their entireties herein.

In some cases, Principal Component Analysis (PCA) whitening may be applied to the image descriptor 104 to reduce dimensions of the image descriptor 104 for fast and accurate indexing.

The adaptive sequence search module 100 of the visual place recognition device 550 may include a processor (not shown). The processor may include a system-on-chip (SOC), a computer processing unit (CPU), a graphical processing unit (GPU), a tensor processing unit (TPU), a neuromorphic chip, a vector accelerator, or any other processing system. The processor is capable of executing an algorithm that causes the processor to perform one or more of the actions, operations, methods, or functions described herein.

The adaptive sequence search module 100 is configured to determine a sequence trajectory of the scene 50 for use in visual place recognition. Specifically, the adaptive sequence search module 100 is configured to determine the final sequence

trajectory 350 of the scene 50 for use in visual place recognition. The adaptive sequence search module 100 is configured to generate a difference matrix 200.

In some cases, a Sobel filter 400 is applied to the difference matrix 200 to obtain a difference saliency matrix 410. The CNN-based discriminator 430 is configured to generate a discriminator output 435, which is used to obtain a confidence score 445 of the final sequence trajectory 350. In some cases, a softmax function 440 is applied to the discriminator output 435 to obtain the confidence score 445 of the final sequence trajectory 350.

Figs. 2A-2O illustrate schematic diagrams of various exemplary steps performed by the adaptive sequence search module 100 shown in Fig. 1 for determining the sequence trajectory of the scene 50 (shown in Fig. 1) for use in visual place recognition.

Fig. 2A depicts a schematic diagram of an example of a cache sequence 110 and a database sequence 120. The cache sequence 110 is a stack having a predefined stack length 111. The predefined stack length 111 of the cache sequence 110 may be adjusted based on desired application attributes.

An image descriptor 104 of an image 102 of the scene 50 is added to the cache sequence 110, such that the image descriptor 104 of the image 102 is added after an image descriptor 104 of a preceding image 102 of the scene 50.

Specifically, the image descriptors 104 of the corresponding plurality of images 102 are added to the cache sequence 110 in the input sequence 105 (shown in Fig. 1).

Further, the image descriptor 104 is removed from the cache sequence 110 and added to the database sequence 120 when a number of successive image descriptors 104 added to the cache sequence 110 after the image descriptor 104 is equal to the predefined stack length 111. In other words, a new image descriptor 104 of a new image 102 of the scene 50 is added at the end of the cache sequence 110 and an oldest image descriptor 104 or the oldest image 102 in the cache sequence 110 is removed from the cache sequence 110 and added to the database sequence 120 to maintain the predefined stack length 111.

For example, in Fig. 2A, the cache sequence 110 includes image descriptors 104a, 104b, 104c, 104d in a sequence in which they were added to the cache sequence 110. Specifically, the image descriptors 104a, 104b, 104c, 104d correspond to respective images 102a, 102b, 102c, 102d of the scene 50. The images 102a, 102b, 102c, 102d of the scene 50 are obtained in the input sequence 105. A new image descriptor 104n corresponding to a new image 102n is to be added to the cache sequence 110.

In the next step, as depicted by Fig. 2B, the new image descriptor 104n is added after the image descriptor 104d of the preceding image 102d of the scene 50.

Further, in the illustrated example of Figs. 2A and 2B, the predefined stack length 111 is four. Therefore, the image descriptor 104a is removed from the cache sequence 110 and added to the database sequence 120 when the number of successive image descriptors 104 (e.g., the image descriptors 104b, 104c, 104d, 104n) added to the cache sequence 110 after the image descriptor 104a is equal to the predefined stack length 111 (i.e., four).

A length of the database sequence 120 may therefore increase with time.

Specifically, the length of the database sequence 120 may increase as more image descriptors 104 corresponding to more images 102 are received with time.

As depicted by Fig. 2C, the adaptive sequence search module 100 (shown in Fig. 1) is configured to determine the difference matrix 200 having a plurality of difference scores 202 by comparing each image descriptor 104 in the cache sequence 110 with each image descriptor 104 in the database sequence 120. Specifically, each of the plurality of difference scores 202 is equal to a difference between the corresponding image descriptor 104 in the cache sequence 110 and the corresponding image descriptor 104 in the database sequence 120. For example, in the illustrated example of Fig. 2C, the difference score 202a is equal to a difference between an image descriptor 104x corresponding to an image 102x in the cache sequence 110 and an image descriptor 104y corresponding to an image 102y in the database sequence 120. The difference matrix 200 is used to describe a dissimilarity between the images 102x, 102y. Specifically, the difference matrix 200 is used to describe a dissimilarity between the image descriptor 104x in the cache sequence 110 and the

image descriptor 104y in the database sequence 120. The difference matrix 200 may be expressed using the Equation 1 provided below.

$$D \in \mathbb{R}^{N \times qds} \quad (1)$$

where,

5 D is the difference matrix 200;

N is the length of the database sequence 120; and

qds is the predefined stack length 111 of the cache sequence 110.

In some cases, the difference matrix 200 is determined after the length of the database sequence 120 is equal to or greater than a predefined database length.

10 Referring to Figs. 2D-2G, the adaptive sequence search module 100 is configured to generate a plurality of blocks. In the illustrated example of Figs. 2D-2G, the adaptive sequence search module 100 is configured to generate a plurality of minimal blocks 222.

In order to generate the plurality of minimal blocks 222, the difference matrix 200 is
 15 sub-sampled by a minimal pooling layer 220. Specifically, the plurality of minimal blocks 222 are generated by sliding the minimal pooling layer 220 having a kernel size 225. The minimal pooling layer 220 has the kernel size 225 equal to the predefined stack length 111 (shown in Fig. 2A). Each of the plurality of minimal blocks 222 has a minimum difference score 224. Further, a stride of the minimal
 20 pooling layer 220 is set equal to the predefined stack length 111. Therefore, in the illustrated example of Figs. 2D-2G, the same kernel size 225 and the stride of the minimal pooling layer 220 is set to four. The minimal pooling layer 220 with the same kernel size 225 and the stride is employed to prevent adjacent blocks from overlapping.

25 The adaptive sequence search module 100 is further configured to select a predetermined number of minimal blocks having the least minimum difference score 224 from the plurality of minimal blocks 222 as a plurality of sequence blocks 210 (shown in Fig. 2H). In some cases, the predetermined number of minimal blocks

may be five. The predetermined number of minimal blocks may be adjusted based on desired application attributes. Therefore, each of the plurality of sequence blocks 210 is a sub-set of the difference matrix 200. The plurality of sequence blocks 210 may be expressed using the Equation 2 provided below.

$$B = \{B^1, \dots, B^j, \dots, B^K\} \quad (2)$$

where,

B is the plurality of sequence blocks 210; and

K is the predetermined number of minimal blocks selected as the plurality of sequence blocks 210.

10 For example, in the illustrated example of Fig. 2H, the adaptive sequence search module 100 selects two of the minimal blocks 222 having the least minimum difference score 224 from the plurality of minimal blocks 222 as the plurality of sequence blocks 210.

The adaptive sequence search module 100 is configured to determine the plurality of
15 sequence blocks 210 from the difference matrix 200 for obtaining a plurality of sequence trajectories 300 (shown in Fig. 2O) corresponding to the plurality of sequence blocks 210. For example, the adaptive sequence search module 100 determines two sequence blocks 210 from the difference matrix 200 for obtaining two sequence trajectories 300.

20 As shown in Fig. 2I, the adaptive sequence search module 100 is further configured to select a difference score 202 having a minimum value in each of the plurality of sequence blocks 210 as a starting point 302 of the corresponding sequence trajectory 300 (shown in Fig. 2O). A plurality of starting points 302 corresponding to the plurality of sequence blocks 210 may be expressed using the Equation 3
25 provided below.

$$s = \{s^1, \dots, s^j, \dots, s^K\} \quad (3)$$

where,

s is the plurality of starting points 302; and

K is a total number of the plurality of starting points 302.

Figs. 2J-2N depict schematic diagrams of various steps performed by the adaptive sequence search module 100 for determining the corresponding sequence trajectory 300 (shown in Fig. 2O) from the starting point 302.

Fig. 2J illustrates one sequence blocks 210 from the plurality of sequence blocks 210. Referring to Fig. 2K, the adaptive sequence search module 100 shown in Fig. 1 is further configured to select a difference score 202 having a minimum value within a predetermined neighbourhood range 310 of a current sequence point 303 (e.g., the starting point 302 in Fig. 2K) in each of the plurality of sequence blocks 210 as a next sequential point 304 of the corresponding sequence trajectory 300. Therefore, the current sequence point 303 is the starting point 302 of the corresponding sequence trajectory 300 when the sequential point 304 next to the starting point 302 is determined.

The predetermined neighbourhood range 310 is located along a column of the corresponding sequence block 210 adjacent to the current sequence point 303. The limited area of the predetermined neighbourhood range 310 may ensure real-time and high efficiency for determining the corresponding sequence trajectory 300.

Further, referring to Fig. 2L, the adaptive sequence search module 100 shown in Fig. 1 is configured to select a difference score 202 having a minimum value within the predetermined neighbourhood range 310 of a current sequence point 303 (e.g., the sequential point 304 of Fig. 2K) in each of the plurality of sequence blocks 210 as a next sequential point 304 of the corresponding sequence trajectory 300. Similarly, the adaptive sequence search module 100 repeats to select a difference score 202 having a minimum value within the predetermined neighbourhood range 310 of a current sequence point 303 in each of the plurality of sequence blocks 210 as a next sequential point 304 of the corresponding sequence trajectory 300 to determine the sequential points 304 of the corresponding sequence trajectory 300 until the plurality of sequence trajectories 300 are obtained.

Referring to Fig. 2M, the adaptive sequence search module 100 is configured to select a difference score 202 having a minimum value within the predetermined neighbourhood range 310 of a current sequence point 303 for opposing sides of the starting point 302 to determine the sequential points 304 on the opposing sides of

the starting point 302 until the corresponding sequence trajectory 300 has a predefined trajectory length 320 as depicted in Fig. 2N. In some cases, the predefined trajectory length 320 is equal to the predefined stack length 111 (shown in Fig. 2A). Therefore, the adaptive sequence search module 100 iteratively selects a difference score 202 having a minimum value within the predetermined neighbourhood range 310 of a current sequence point 303 in each of the plurality of sequence blocks 210 to obtain the corresponding sequence trajectory 300.

For example, the predetermined neighbourhood range 310 is located along a column of the corresponding sequence block 210 on the left side of the starting point 302 in Figs. 2K and 2L and the predetermined neighbourhood range 310 is located along a column of the corresponding sequence block 210 on the right side of the starting point 302 in Fig. 2M.

Further, referring to Fig. 2O, the adaptive sequence search module 100 (shown in Fig. 1) is configured to select a final sequence trajectory 350 from the plurality of sequence trajectories 300. In some cases, a sum of the difference scores 202 of the starting point 302 and the sequential points 304 of the corresponding sequence trajectory 300 is determined for each of the plurality of sequence trajectories 300. Further, a sequence trajectory 300 having a minimum value of the sum of the difference scores 202 is selected from the plurality of sequence trajectories 300 as the final sequence trajectory 350. The final sequence trajectory 350 may be expressed using the Equation 4 provided below.

$$i_{rs} = \text{argmin} (\text{difference scores}) \quad (4)$$

where i_{rs} is the final sequence trajectory 350.

Fig. 3 illustrates a schematic diagram of an exemplary step performed by the Sobel filter 400 shown in Fig. 1 for obtaining the difference saliency matrix 410. As discussed above, the Sobel filter 400 is applied to the difference matrix 200 to obtain the difference saliency matrix 410. The Sobel filter 400 is a discrete differentiation operator which is typically used to find a gradient of an image (e.g., the image 102). The difference saliency matrix 410 may be expressed using the Equation 5 provided below.

$$Ds = \text{Sobel} (D) \quad (5)$$

where,

D is the difference matrix 200; and
 D_s is the difference saliency matrix 410.

Figs. 4A-4C illustrate schematic diagrams of exemplary steps to obtain a concatenated matrix 420. In some cases, a portion 205 of the difference matrix 200 (shown in Fig. 3) corresponding to the final sequence trajectory 350 and a portion 415 of the difference saliency matrix 410 (shown in Fig. 3) corresponding to the final sequence trajectory 350 are concatenated to form the concatenated matrix 420.

The portion 205 of the difference matrix 200 corresponding to the final sequence trajectory 350 and the portion 415 of the difference saliency matrix 410 corresponding to the final sequence trajectory 350 may be expressed using the Equation 6 provided below.

$$D^{irs} = D[w_s:w_e], D_s^{irs} = D_s[w_s:w_e] \quad (6)$$

where,

D^{irs} is the portion 205 of the difference matrix 200 corresponding to the final sequence trajectory 350;

D_s^{irs} is the portion 415 of the difference saliency matrix 410 corresponding to the final sequence trajectory 350;

$w_s = \min(\text{coordinates of the final sequence trajectory}) - W_{\text{range}}$; and

$w_e = \max(\text{coordinates of the final sequence trajectory}) + W_{\text{range}}$.

In the illustrated example of Figs. 4A-4C, W_{range} is equal to two. However, W_{range} may be selected as per desired application attributes.

Fig. 5 illustrates a schematic block diagram of an example architecture of the CNN-based discriminator 430. The CNN-based discriminator 430 is applied to the concatenated matrix 420 to generate the discriminator output 435. The CNN-based discriminator 430 has a multi-layer CNN model. The CNN-based discriminator 430 includes a series of a CNN layer 450 having a kernel size of 3x3, a batch normalization layer 451, a rectified linear activation function (ReLU) layer 452, and a max pooling layer 453 having a kernel size of 2x2. An output of the max pooling layer 453 is provided to a block including a series of a fully connected (FC) layer 454 and another ReLU layer 455. The block is repeated, and an output of the block is provided to another FC layer 456 to generate the discriminator output 435.

Further, the softmax function 440 is applied to the discriminator output 435 to obtain the confidence score 445 of the final sequence trajectory 350 (shown in Fig. 2O).

The confidence score 445 of the final sequence trajectory 350 may be expressed using the Equation 7 provided below.

$$S_{\text{conf}} = \text{Softmax}(\text{DD}_s\text{Net}(\text{DD}_s)) \quad (7)$$

where,

S_{conf} is the confidence score 445;

$\text{DD}_s\text{Net}(\text{DD}_s)$ is the discriminator output 435; and

DD_s is the concatenated matrix 420.

Therefore, the CNN-based discriminator 430 is used to validate if the final sequence trajectory 350 is a true correspondence. By learning the optimal judgmental inference from dataset statistics in a data-driven manner, the CNN-based discriminator 430 may mitigate false positives caused by artificial rule and threshold. In other words, the CNN-based discriminator 430 may evaluate confidence of the final sequence trajectory 350 efficiently and accurately and may significantly alleviate the false-positive and the false-negative retrievals.

Fig. 6 illustrates an example flow chart for a computer implemented method 500 for determining the sequence trajectory of the scene 50 for use in visual place recognition. The method 500 will be described with reference to Figs. 2A-2O, 3, 4A-4C, 5, and 6.

At step 502, the method 500 includes determining the difference matrix 200 having the plurality of difference scores 202 by comparing each image descriptor 104 in the cache sequence 110 with each image descriptor 104 in the database sequence 120. As discussed above, each image descriptor 104 corresponds to a respective image 102 of the scene 50.

The image descriptor 104 of the image 102 of the scene 50 is added to the cache sequence 110, such that the image descriptor 104 of the image 102 is added after the image descriptor 104 of the preceding image 102 of the scene 50. Further, the image descriptor 104 is removed from the cache sequence 110 and added to the database sequence 120 when the number of successive image descriptors 104

added to the cache sequence 110 after the image descriptor 104 is equal to the predefined stack length 111.

In step 502, each of the plurality of difference scores 202 is equal to the difference between the corresponding image descriptor 104 in the cache sequence 110 and the
5 corresponding image descriptor 104 in the database sequence 120.

At step 504, the method 500 includes determining the plurality of sequence blocks 210 from the difference matrix 200 for obtaining the plurality of sequence trajectories 300 corresponding to the plurality of sequence blocks 210. Each of the plurality of sequence blocks 210 is the sub-set of the difference matrix 200.

10 In step 504, the difference matrix 200 is sub-sampled by the minimal pooling layer 220 to generate the plurality of minimal blocks 222. Each of the plurality of minimal blocks 222 has the minimum difference score 224. The predetermined number of minimal blocks having the least minimum difference score 224 is selected from the plurality of minimal blocks 222 as the plurality of sequence blocks 210. The minimal
15 pooling layer 220 has the kernel size 225 equal to the predefined stack length 111.

At step 506, the method 500 includes selecting the difference score 202 having the minimum value in each of the plurality of sequence blocks 210 as the starting point 302 of the corresponding sequence trajectory 300.

At step 508, the method 500 includes selecting the difference score 202 having the
20 minimum value within the predetermined neighbourhood range 310 of the current sequence point 303 in each of the plurality of sequence blocks 210 as the next sequential point 304 of the corresponding sequence trajectory 300. The current sequence point 303 is the starting point 302 of the corresponding sequence trajectory 300 when the sequential point 304 next to the starting point 302 is
25 determined.

The predetermined neighbourhood range 310 is located along the column of the corresponding sequence block 210 adjacent to the current sequence point 303.

At step 510, the method 500 includes repeating the step 508 to determine the sequential points 304 of the corresponding sequence trajectory 300 until the plurality of sequence trajectories 300 are obtained.

The steps 508 to 510 are repeated for opposing sides of the starting point 302 to
5 determine the sequential points 304 on the opposing sides of the starting point 302 until the corresponding sequence trajectory 300 has the predefined trajectory length 320. The cache sequence 110 is the stack having the predefined stack length 111. Further, the predefined trajectory length 320 is equal to the predefined stack length 111.

10 At step 512, the method 500 includes selecting the final sequence trajectory 350 from the plurality of sequence trajectories 300.

In step 512, the sum of the difference scores 202 of the starting point 302 and the sequential points 304 of the corresponding sequence trajectory 300 is determined for each of the plurality of sequence trajectories 300. The sequence trajectory 300
15 having the minimum value of the sum of the difference scores 202 is selected from the plurality of sequence trajectories 300 as the final sequence trajectory 350.

In some cases, the Sobel filter 400 is applied to the difference matrix 200 to obtain the difference saliency matrix 410. Further, the portion 205 of the difference matrix 200 corresponding to the final sequence trajectory 350 and the portion 415 of the
20 difference saliency matrix 410 corresponding to the final sequence trajectory 350 are concatenated to form the concatenated matrix 420. The CNN-based discriminator 430 is applied to the concatenated matrix 420 to generate the discriminator output 435.

Further, the softmax function 440 is applied to the discriminator output 435 to obtain
25 the confidence score 445 of the final sequence trajectory 350.

A method for visual place recognition is also provided. The method includes obtaining the plurality of images 102 of the scene 50 in the input sequence 105. The method further includes performing the method 500 on the plurality of images 102 to obtain the final sequence trajectory 350 of the scene 50. The method further

includes generating the control signal 555 that causes the device 600 to perform the action based on the final sequence trajectory 350.

A computer program, a machine-readable storage medium, or a data signal is also provided that includes instructions such that, upon execution on a data processing device, cause the device to perform one, some, or all of the steps of a method disclosed herein.

Various experiments on city-scale benchmark datasets were conducted, namely Dataset 1 (d2d), Dataset 1 (d2n), Dataset 2 (d2d), and Dataset 2 (d2n), which demonstrated that the method 500 according to the present invention outperforms existing methods and achieves a significantly better performance on all datasets. 'd2d' denotes 'day-to-day' and 'd2n' denotes 'day-to-night'.

The results of the experiments are summarized in the Table 1 provided below.

Table 1

VPR Pipeline	Precision (%)			
	Dataset 1 (d2d)	Dataset 1 (d2n)	Dataset 2 (d2d)	Dataset 2 (d2n)
NetVLAD	52.4	32.6	81.3	44.0
NetVLAD + SeqSLAM	57.9	34.1	87.5	50.5
NetVLAD + AdaptSeq	59.0	38.7	88.7	56.0
APPSVR	65.2	47.1	88.0	57.6
APPSVR + SeqSLAM	70.0	52.6	92.3	67.9
APPSVR +	74.1	55.7	93.6	69.7

AdaptSeq				
----------	--	--	--	--

In the first row of the Table 1, the precision of VPR pipeline obtained using NetVLAD (Vector of Locally Aggregated Descriptors) for Dataset 1 (d2d), Dataset 1 (d2n), Dataset 2 (d2d), and Dataset 2 (d2n) is listed. Specifically, NetVLAD was used to obtain the image descriptors.

In the second row of the Table 1, the precision of VPR pipeline obtained using NetVLAD in combination with SeqSLAM (Simultaneous Localization and Mapping) for Dataset 1 (d2d), Dataset 1 (d2n), Dataset 2 (d2d), and Dataset 2 (d2n) is listed. Specifically, NetVLAD was used to obtain the image descriptors and SeqSLAM was used for determining sequence trajectory.

In the third row of the Table 1, the precision percentages of VPR pipeline obtained using NetVLAD in combination with AdaptSeq (i.e., the method 500 of the present invention shown in Fig. 6) for Dataset 1 (d2d), Dataset 1 (d2n), Dataset 2 (d2d), and Dataset 2 (d2n) is listed. Specifically, NetVLAD was used to obtain the image descriptors and AdaptSeq was used for determining sequence trajectory.

In the fourth row of the Table 1, the precision percentages of VPR pipeline obtained using Attentional Pyramid Pooling of Salient Visual Residuals (APPSVR) for Dataset 1 (d2d), Dataset 1 (d2n), Dataset 2 (d2d), and Dataset 2 (d2n) is listed. Specifically, APPSVR was used to obtain the image descriptors. APPSVR is described in WO 2022/258624 A1 and G. Peng, J. Zhang, H. Li and D. Wang, "Attentional Pyramid Pooling of Salient Visual Residuals for Place Recognition," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 865-874, doi:10.1109/ICCV48922.2021.00092.

In the fifth row of the Table 1, the precision percentages of VPR pipeline obtained using APPSVR in combination with SeqSLAM for Dataset 1 (d2d), Dataset 1 (d2n), Dataset 2 (d2d), and Dataset 2 (d2n) is listed. Specifically, APPSVR was used to obtain the image descriptors and SeqSLAM was used for determining sequence trajectory.

In the sixth row of the Table 1, the precision percentages of VPR pipeline obtained using APPSVR in combination with AdaptSeq (i.e., the method 500 of the present invention shown in Fig. 6) for Dataset 1 (d2d), Dataset 1 (d2n), Dataset 2 (d2d), and Dataset 2 (d2n) is listed. Specifically, APPSVR was used to obtain the image
 5 descriptors and AdaptSeq was used for determining sequence trajectory.

As is apparent from Table 1, AdaptSeq outperformed the modified SeqSLAM (see rows 2, 3, 5, and 6). Further, APPSVR in combination with AdaptSeq achieved the best results (see row 6). Compared to the baseline NetVLAD + SeqSLAM, stable advantages of 16.2%, 21.6%, 6.1%, and 19.2% can be observed on the respective
 10 Dataset 1 (d2d), Dataset 1 (d2n), Dataset 2 (d2d), and Dataset 2 (d2n)). Moreover, the performance boost of over 20% is more pronounced on Dataset 1 (d2n) and Dataset 2 (d2n) which are night datasets. This further demonstrates robustness of the proposed VPR pipeline, i.e., APPSVR in combination with AdaptSeq.

Figs. 7A-7B depict example sequence trajectories 702, 704, respectively.

Specifically, the sequence trajectory 702 of Fig. 7A is obtained using AdaptSeq without verification by the CNN-based discriminator 430 shown in Fig. 1. Further, the sequence trajectory 704 of Fig. 7B is obtained using AdaptSeq with verification by the CNN-based discriminator 430. The darker portions of the sequence
 15 trajectories 702, 704 depict incorrect retrievals, whereas lighter portions of the sequence trajectories 702, 704 depict correct retrievals. It can be observed that by introducing the CNN-based discriminator 430, the VPR pipeline APPSVR in combination with AdaptSeq is able to reduce the number of false positives significantly while keeping as many true positives as possible.
 20

Fig. 8 illustrates an example plot 800 depicting various precision versus recall curves of Dataset 2 (d2d) and Dataset 2 (d2n) using AdaptSeq with and without verification by the CNN-based discriminator 430 shown in Fig. 1. Recall is expressed in the
 25 abscissa. Precision is expressed in the ordinate.

Specifically, the plot 800 includes a curve 802 which represents a precision-recall curve of Dataset 2 (d2d) using AdaptSeq without verification by the CNN-based
 30 discriminator 430.

Further, the plot 800 includes a curve 804 which represents a precision-recall curve of Dataset 2 (d2n) using AdaptSeq without verification by the CNN-based discriminator 430.

5 The plot 800 further includes a curve 806 which represents a precision-recall curve of Dataset 2 (d2d) using AdaptSeq with verification by the CNN-based discriminator 430.

The plot 800 further includes a curve 808 which represents a precision-recall curve of Dataset 2 (d2n) using AdaptSeq with verification by the CNN-based discriminator 430.

10 As is apparent from the curves 802, 804, 806, 808, the precision of AdaptSeq with verification by the CNN-based discriminator 430 is always higher than the precision obtained without verification by the CNN-based discriminator 430 on both datasets (i.e., Dataset 2 (d2d) and Dataset 2 (d2n)). This verifies the effectiveness of the CNN-based discriminator 430 for VPR.

15 It will be understood that the invention is not limited to the embodiments above-described and various modifications and improvements can be made without departing from the concepts described herein. Except where mutually exclusive, any of the features may be employed separately or in combination with any other features and the disclosure extends to and includes all combinations and sub-
20 combinations of one or more features described herein.

REFERENCE SIGNS

	10	Encoder
	50	Scene
	100	Adaptive Sequence Search Module
5	102	Image
	102a	Image
	102b	Image
	102c	Image
	102d	Image
10	102n	Image
	102x	Image
	102y	Image
	104	Image Descriptor
	104a	Image Descriptor
15	104b	Image Descriptor
	104c	Image Descriptor
	104d	Image Descriptor
	104n	Image Descriptor
	104x	Image Descriptor
20	104y	Image Descriptor
	105	Input Sequence
	110	Cache Sequence
	111	Predefined Stack Length
	120	Database Sequence
25	200	Difference Matrix
	202	Difference Scores
	202a	Difference Score
	205	Portion of Difference Matrix
	210	Sequence Blocks
30	220	Minimal Pooling Layer
	222	Minimal Blocks
	224	Minimum Difference Score
	225	Kernel Size
	300	Sequence Trajectories
35	302	Starting Point
	303	Current Sequential Point
	304	Next Sequential Point
	310	Predetermined Neighbourhood Range
	320	Predefined Trajectory Length
40	350	Final Sequence Trajectory
	400	Sobel Filter
	410	Difference Saliency Matrix
	415	Portion of Difference Saliency Matrix

	420	Concatenated Matrix
	430	Convolutional Neural Network-Based Discriminator
	435	Discriminator Output
	440	Softmax Function
5	445	Confidence Score
	450	CNN Layer
	451	Batch Normalization Layer
	452	Rectified Linear Activation Function (ReLU) Layer
	453	Max Pooling Layer
10	454	Fully Connected (FC) Layer
	455	ReLU layer
	456	FC Layer
	500	Method
	502	Step
15	504	Step
	506	Step
	508	Step
	510	Step
	512	Step
20	550	Visual Place Recognition Device
	555	Control Signal
	600	Device
	702	Sequence Trajectory
	704	Sequence Trajectory
25	800	Plot
	802	Curve
	804	Curve
	806	Curve
	808	Curve
30	W_{range}	Range

CLAIMS

1. A computer implemented method (500) for determining a sequence trajectory of a scene (50) for use in visual place recognition, the method (500) comprising:
 - a) determining a difference matrix (200) having a plurality of difference scores (202) by comparing each image descriptor (104) in a cache sequence (110) with each image descriptor (104) in a database sequence (120), each image descriptor (104) corresponding to a respective image (102) of the scene (50);
 - b) determining a plurality of sequence blocks (210) from the difference matrix (200) for obtaining a plurality of sequence trajectories (300) corresponding to the plurality of sequence blocks (210), each of the plurality of sequence blocks (210) being a sub-set of the difference matrix (200);
 - c) selecting a difference score (202) having a minimum value in each of the plurality of sequence blocks (210) as a starting point (302) of the corresponding sequence trajectory (300);
 - d) selecting a difference score (202) having a minimum value within a predetermined neighbourhood range (310) of a current sequence point (303) in each of the plurality of sequence blocks (210) as a next sequential point (304) of the corresponding sequence trajectory (300), the current sequence point (303) being the starting point (302) of the corresponding sequence trajectory (300) when the sequential point (304) next to the starting point (302) is determined;
 - e) repeating step d) to determine the sequential points (304) of the corresponding sequence trajectory (300) until the plurality of sequence trajectories (300) are obtained; and
 - f) selecting a final sequence trajectory (350) from the plurality of sequence trajectories (300).
2. The method (500) according to claim 1, characterized in that, steps d) to e) are repeated for opposing sides of the starting point (302) to determine the sequential points (304) on the opposing sides of the starting point (302) until the

corresponding sequence trajectory (300) has a predefined trajectory length (320).

3. The method (500) according to claim 2, characterized in that, in step b) the difference matrix (200) is sub-sampled by a minimal pooling layer (220) to generate a plurality of minimal blocks (222), each of the plurality of minimal blocks (222) having a minimum difference score (224), wherein a predetermined number of minimal blocks having the least minimum difference score (224) is selected from the plurality of minimal blocks (222) as the plurality of sequence blocks (210).
4. The method (500) according to claim 3, characterized in that, the cache sequence (110) is a stack having a predefined stack length (111), wherein the predefined trajectory length (320) is equal to the predefined stack length (111), and preferably the minimal pooling layer (220) has a kernel size (225) equal to the predefined stack length (111).
5. The method (500) according to claim 4, characterized in that, an image descriptor (104) of an image (102) of the scene (50) is added to the cache sequence (110), such that the image descriptor (104) of the image (102) is added after an image descriptor (104) of a preceding image (102) of the scene (50).
6. The method (500) according to claim 5, characterized in that, the image descriptor (104) is removed from the cache sequence (110) and added to the database sequence (120) when a number of successive image descriptors (104) added to the cache sequence (110) after the image descriptor (104) is equal to the predefined stack length (111).
7. The method (500) according to any of the preceding claims, characterized in that, the predetermined neighbourhood range (310) is located along a column of the corresponding sequence block (210) adjacent to the current sequence point (303).

8. The method (500) according to any of the preceding claims, characterized in that, in step f) a sum of the difference scores (202) of the starting point (302) and the sequential points (304) of the corresponding sequence trajectory (300) is determined for each of the plurality of sequence trajectories (300), wherein a sequence trajectory (300) having a minimum value of the sum of the difference scores (202) is selected from the plurality of sequence trajectories (300) as the final sequence trajectory (350).
9. The method (500) according to any of the preceding claims, characterized in that, a Sobel filter (400) is applied to the difference matrix (200) to obtain a difference saliency matrix (410).
10. The method (500) according to claim 9, characterized in that, a portion (205) of the difference matrix (200) corresponding to the final sequence trajectory (350) and a portion (415) of the difference saliency matrix (410) corresponding to the final sequence trajectory (350) are concatenated to form a concatenated matrix (420), wherein a convolutional neural network-based discriminator (430) is applied to the concatenated matrix (420) to generate a discriminator output (435).
11. The method (500) according to claim 10, characterized in that, a softmax function (440) is applied to the discriminator output (435) to obtain a confidence score (445) of the final sequence trajectory (350).
12. The method (500) according to any of the preceding claims, characterized in that, in step a) each of the plurality of difference scores (202) is equal to a difference between the corresponding image descriptor (104) in the cache sequence (110) and the corresponding image descriptor (104) in the database sequence (120).
13. A method for visual place recognition, the method comprising:

- 1) obtaining a plurality of images (102) of a scene (50) in an input sequence (105);
- 2) performing the method (500) according to any of the preceding claims on the plurality of images (102) to obtain a final sequence trajectory (350) of the scene (50); and
- 3) generating a control signal (555) that causes a device (600) to perform an action based on the final sequence trajectory (350).

14. A vehicle comprising:

an image sensor configured to obtain a plurality of images (102) of a scene (50);

a visual place recognition device (550) that is configured to perform the method according to claim 13, so as to obtain a control signal (555); and

a control unit (600) that is configured to process the control signal (555) and perform an action based on the control signal (555).

15. A computer program, a machine-readable storage medium, or a data signal that comprises instructions that, upon execution on a data processing device, cause the device to perform one, some, or all of the steps of the method according to any of the preceding claims 1 to 13.