This is an Accepted Manuscript. You may cite and find the definitive published version as following: K. Fathi, A. Darvishy and H. W. van de Venn, "Similarity and Location-based Real-time Loop Closure: SNAPS for SLAM† in Unexplored-Environments," 2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS), Berlin, Germany, 2023, pp. 1-7, doi: 10.1109/COINS57856.2023.10189300.

Similarity and Location-based Real-time Loop Closure: SNAPS for SLAM† in Unexplored Environments

***Abstract*—Loop closure is an inseparable part of any accurate and reliable visual simultaneous localization and mapping (SLAM) algorithm for autonomous vehicles and mobile robots. Loop closure potentially decreases the impact of the cumulative drift while generating the map of the traversed environment. In this paper, a heuristic similarity and location-based approach for loop closure in unexplored environments is introduced. The current SLAM implementation on average requires 0.295 seconds per frame from which only 0.0270 seconds are the runtime latencies of the similarity and location-based real-time loop closure (SNAPS), which includes trajectory correction. The proposed approach results in a 65% decrease in the mean deviation from the ground truth. In the conducted study, neither conventional bag-of-words models, nor computationally expensive deep neural networks have been used to detect and perform loop closure, which makes the proposed approach both interpretable and efficient. In fact, we propose a method which tries to find loop closure candidates based on the location and also an interpretable similarity score attained from the generated thumbnails of the read frames instead of the local descriptors. Additionally, the employed discount factor applied on the pose trajectory update rule guarantees a consistent and accurate map. Lastly, the KITTI dataset is used to demonstrate the efficiency and accuracy of SNAPS for SLAM.**

***Index Terms*—Visual simultaneous localization and mapping, Visual odometry, Loop closure detection, Localization, Autonomous vehicles**

I. Introduction

One of the most important topics in autonomous vehicles and mobile robots is simultaneous localization and mapping (SLAM), which helps the intelligent system autonomously navigate in unknown environments by generating a map of the traversed environment. Additionally, SLAM allows the intelligent system to localize itself on the calculated map. There have been various implementations of SLAM employing different sensors such as rotary encoders, inertial measurement units, Global Positioning System, laser range sensors [1]–[4]. However, in this paper, we focus on visual SLAM (vSLAM) which uses a camera as its main source of information.

Regardless of the deployed vSLAM method, while performing visual odometry, there is an inevitable drift in calculating the trajectory of autonomous mobile systems describing the pose of the system through time, impacting the estimated map of the environment. Thus, to acquire a consistent and accurate map, the intelligent system must be able to recognize formerly visited places and create data associations between the former points in the calculated trajectory and the revisited locations. This association is referred to as loop closure. Recent studies, address both traditional and deep learning-based methods for performing loop closure. As pointed out by Xia et al. [5], the former tries to generate hand-crafted features to create a bag-of-words with either an online or offline vocabulary for a given environment. These hand-crafted features could be Fisher vector [6], Vector of Locally Aggregated Descriptors [7], Scale-Invariant Feature Transform (SIFT) [8], Speeded-Up Robust Features (SURF) [9] or Oriented FAST and Rotated BRIEF (ORB) [10].

The research by Arshad and Kim [11] has compared the performance of the online and offline vocabularies and states that the offline vocabulary cannot perform well if the aforementioned vocabulary is gathered from a different scene than the test environment. This characteristic reduces the generalization capability of these methods for unexplored environments.

Furthermore, researchers such as Arshad and Kim [11], Shin and Ho [12] and Naseer et al. [13], suggest that one effective way to deal with the downsides of relying on offline vocabularies is to employ deep neural networks.

These deep learning-based methods are essentially trying to reduce the need for hand-crafted features in creating references for comparison. As an example, Gao and Zhang [14] have used stacked auto-encoders to acquire feature representations which is namely the latent space in this architecture. Thereafter, a variant of difference between two arbitrary scenes' feature representation is used to determine whether loop closure should be performed or not. The authors mention, that when the difference is "large", the scenes are not considered to be similar. However, they did not provide any insight or method on how to assign an appropriate threshold as the design parameter of this model. In addition, they merely focused on the precision and recall of the deep neural network, which were 70% and 50% respectively, and did not try to optimize their method to reduce the computational costs.

Additionally, in another work by Zhang et al. [15], they have used pre-trained deep convolutional neural networks, with 25 layers trained with 1.2 million images of 1000 categories, to extract features which were then used to detect loops. Similar to the previous example, they have calculated the distance between feature vectors of different scenes and acquired the similarity scores, making nearby images candidates for loop closure with no discussion over efficiency.

Furthermore, Xia et al. [5] have claimed that hand-crafted features which are designed based on human expertise could potentially be non-representative. Therefore, they have compared different architectures, e.g., AlexNet [16], CaffeNet [17], GoogLeNet [18], etc., for loop closure via classification. In order to train their architectures, they have used the information in their dataset as ground truth to label which images in

† SLAM stands for Simultaneous Localization and Mapping

fact form a loop closure. Moreover, they required 39 GB of RAM memory and an NVIDIA GTX 780 GPU in their studies. This supervised training approach limits the use cases of this model to scenes with available ground truth labels. In fact, this approach merely replaces the offline vocabulary with a deep neural network that tries to encode frame information in the dataset which are potentially forming a loop. In summary, the proposed method is computationally expensive and the authors do not offer solutions for making their model more efficient and less data-demanding.

One important note here is that, by employing deep neural networks, the trained model will be less interpretable compared to classic vision methods; the acquired similarity from the model's feature vectors is also counter-intuitive, as a human operator will not have a good understanding of how these black box models have calculated different feature vectors. Moreover, as pointed out by Huang et al. [19], deep neural networks are susceptible to adversarial attacks which can potentially deteriorate their performance given very small but targeted changes in the input image. Furthermore, Djolonga et al. suggest, that these models could potentially become more robust against data distribution shifts, which is equivalent to new unseen scenarios, by expanding their training datasets [20]. However, one of the initial reasons to use deep neural networks for loop closure was to make the loop closure algorithm independent of some previously acquired offline vocabulary or large training datasets, which are not solved by the proposed solutions.

Additionally, studies targeting traditional methods in loop detection suggest that local descriptors are not robust towards all scenarios, and they leave out the entire image level details [15], [21]. However, the convolutional layers in deep neural networks could be a solution to this problem with the cost of the above-mentioned drawbacks.

Therefore, to deal with the problem of local descriptors and lack of interpretability in deep neural networks, in this paper we have employed a hybrid similarity scoring function introduced by Wang and Bovik [22] called "universal image quality index", which provides a better estimate of the similarity between different scenes. The proposed similarity measurement approach is based on calculating any distortions due to loss of correlation, illumination distortion and contrast distortion. As offline vocabularies have shown to be suboptimal, in the conducted study we have generated low dimensional thumbnails (in our case 1/6 of the original image size) and stored them as representatives of the scenes as the intelligent system is traversing the environment. The similarity calculations are then performed on the aforementioned thumbnails. Our similarity estimation approach has the following advantageous:

1) Keeps the entire image so that higher level details will not be ignored as by the local descriptors
2) Returns an interpretable value as the similarity, consisting of correlation and distortion values, which can be used in later steps of loop closure
3) Reduces the computational costs as the image size has been reduced

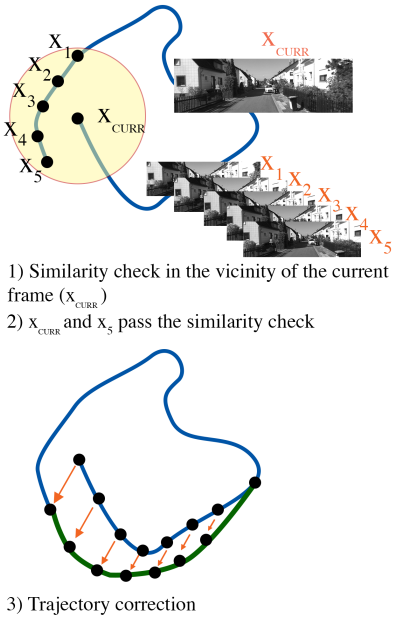

Fig. 1. Overview of SNAPS for SLAM

We have used these interpretable calculated similarity values in the trajectory correction while performing loop closure. Our proposed method is inspired by the idea of discount factor in reinforcement learning which determines the importance of long-term consequences of action [23].

The actions in our case are the calculated trajectory of poses through time. Given the cumulative drift in the calculated trajectory, the initial entries of a trajectory are more reliable than the later ones. We use this heuristic to correct the acquired trajectory upon detecting a loop.

In fact, once the current scene matches an arbitrary reference scene, the currently calculated pose and some previous poses are updated in a way which makes them more resemble the pose at the reference scene. Numerous parameters determine the degree of changes, a.k.a. discount factor, in updating the previously calculated poses which include index difference, inter-frame similarity, etc. In addition, to reduce the computational cost we aimed to reduce the number of times we calculate similarity values by limiting the reference frames to frames which are in the vicinity of the current frame location (Fig. 1).

Our heuristic approach was tested on the KITTI dataset [24], and the results show the efficiency and accuracy of our loop closure technique. The current implementation on average requires 0.295 seconds per frame from which only 0.0270 seconds are the runtime latencies of the similarity and location-based real-time loop closure (SNAPS), including loop closure and trajectory correction. It should be pointed out that the visual odometry approach in this paper is not novel and is not the main focus of the paper. It is possible to substitute the visual odometry part with any desired model.

The main contributions of our proposed method are the

following:

1) Does not require any training datasets, offline vocabularies or computationally expensive architectures
2) The loop closure algorithm can run in real-time and when needed provides a reliable initial starting point for other optimization schemas such as Bundle Adjustment
3) The employed similarity measurement is interpretable and can be used in adjusting the pose trajectory for a more consistent map
4) The discount factor along with the heuristic used, reduces the computational burden and makes our model more efficient

The remainder of this paper is outlined as follows. First, the preliminaries of the visual odometry and the SNAPS for SLAM are given. Afterwards, the implementation details of the proposed method are discussed in detail and lastly, conclusions are made and the future work of the conducted study is introduced.

II. Preliminaries

A. Testbed

The used dataset in the conducted study is gathered from a stereo camera mounted on the roof of a car driving through urban areas [24]. The images are recorded with sampling time of 0.1 seconds and the calibration data for the cameras are provided in a separate file.

B. Disparity Map

By having data from a stereo camera, it is possible to calculate the disparity map and afterwards the depth map of the captured frame. Given the distance between 2 points in the left and right image planes ($x \; and \; x^{'}$), the focal length of the camera (f) and the horizontal distance between the left and right lenses (B), the depth (z) can be calculated as follows:

$$z = \frac{Bf}{x - x^{'}} \tag{1}$$

For acquiring the depth map, we first use the block matching algorithm in Open Source Computer Vision Library (OpenCV) to calculate the disparity map and thereafter apply the disparity map filter based on the Weighted Least Squares filter [25], and finally convert the disparity map to a depth map.

C. Feature extraction and matching for motion estimation

We have employed feature-based visual odometry in our model to show that even without a state-of-the-art visual odometry model, it is possible to remove the drift from the visual odometry with our novel loop closure approach. Furthermore, the SIFT class from OpenCV library is used to perform the following steps for feature extraction and matching from consecutive frames, which can be used to estimate the movement of the camera [26]:

1) Scale-space extrema detection
2) Keypoint localization
3) Orientation assignment
4) Keypoint descriptor generation
5) Keypoint matching

Thereafter, by using the matching keypoints and the corresponding depth values from the depth map, the transformation between two consecutive frames in the world coordinate can be calculated as follows [27]:

$$\begin{aligned} x &= z\frac{u - u_0}{f_0} \\ y &= z\frac{v - v_0}{f_0} \end{aligned} \tag{2}$$

where u and v describe the keypoint position in the image plane, u_0 and v_0 are the offsets of the principal point from the top-left corner of the image plane and f_0 is the focal length.

D. Perspective-n-Point (PnP)

The PnP problem deals with determining the pose of a calibrated camera given n three-dimensional points in the world and the corresponding two-dimensional projections in consecutive frames. Formally, the problem definition for PnP is as follows [27]:

$$sp_c = K[R|T]p_w \tag{3}$$

where p_w is the homogeneous coordinates of a point in the world frame, explained in section II-C, and p_c is the corresponding homogeneous coordinate of the point in the image plane. Additionally, K is the calibration matrix of intrinsic camera parameters. Lastly, R and T are the rotation matrix and translation vector which are required. Moreover, we have used Random sample consensus (RANSAC) algorithm [28], to deal with outliers and find the optimal solution for the PnP problem. Having the transformations and the pose, now we can create a trajectory of poses of the camera and refine it upon detecting a loop.

E. Universal Image Quality Index

Universal image quality index can be used to model any image distortions caused by loss of correlation, luminance distortion and contrast distortion. Let $x = \{x_i : i = 1, 2, \cdots, N\}$ and $y = \{y_i : i = 1, 2, \cdots, N\}$ be two images. The index is defined as [22]:

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]} \tag{4}$$

where

$$\begin{aligned} \bar{x} &= \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \\ \sigma_x^2 &= \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \\ \sigma_{xy} &= \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}). \end{aligned} \tag{5}$$

Moreover, this index takes the degree of linear correlation between x and y into account along with how close the

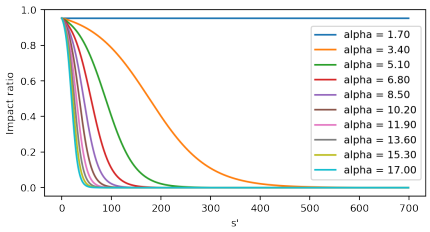

Fig. 2. Impact of number of found loops (α) on the trajectory correction

mean luminance and the contrast of the images x and y are. Similar to convolutional layers, this index starts from the top-left corner of an image and slides a window of size $B \times B$ vertically and horizontally throughout the image.

III. Implemented Method

A. Thumbnail generation

While reading the image frames, we generate thumbnails of the read images which are essentially the rescaled images to a lower dimension (shown as $x_1, x_2, \cdots, x_5$ in Fig. 1). The thumbnails are used to calculate how similar two arbitrary frames are. In the conducted study we use $1/6$ as our rescaling factor. In addition, we generate the thumbnails online as the frames are read, and do not require any vocabularies or trained models which increases the generalization power of our approach.

B. Finding loop closure candidates

As the algorithm is reading frames and calculating the pose trajectory from the original images, the generated thumbnails are used to check whether the currently read frame match any of the formerly seen frames. Moreover, to reduce the computational costs, we have designated an allowed vicinity around every read frame, where the similarity checks are performed. The aforementioned vicinity is depicted with a yellow circle in Fig. 1.

As the thumbnails in the vicinity of the current frame are checked for similarity, a designated counter, tries to control the number of attempts for finding a similar frame in the vicinity of the current frame. If the number of attempts exceeds a predefined value, the process of finding a loop closure candidate stops and the pose for the next upcoming frame is calculated.

Additionally, if the currently read frame is the N^{th} frame in the trajectory, the last M frames will not be considered as candidates for loop closure. We call this parameter M the $frame\ distance$. This design parameter also further reduces the computational costs of finding candidates for loop closure. Moreover, to avoid the attempt for finding a reference frame in scenarios such as junctions where the car or intelligent system is entering the formerly seen frame from a different road, we also check the yaw angle between the current pose and the reference frame pose and if the difference is bigger than a predefined value (in our case 10 degrees), we ignore the reference frame.

After all the previous checks, if the similarity between the frames in the vicinity of the current frame reach a given threshold (in our case set to 75%), this frame will be used as a reference to correct the last K frames in the pose trajectory, resulting in a more consistent map. Unlike other methods employed in recent studies, in this paper, the factors affecting the calculated similarity index are interpretable and easy to understand for a human operator. Thus, it is quite convenient to fine-tune this parameter if needed.

C. Closing the loop and updating the trajectory

Once a reference frame passes all the checks for similarity and distance to the current frame, the position is updated as follows:

$$
\begin{aligned}
&\delta_x = x_{ref} - x_{curr}, \quad \delta_y = y_{ref} - y_{curr}, \quad \delta_z = z_{ref} - z_{curr} \\
&For\ s \in [N-K, N]: \\
&\quad s^{'} = N - s \\
&\quad \gamma = (1 - \frac{1}{e^{3-\alpha*0.01*s'}}) \\
&x_s^{new} = Q.\gamma.\delta_x, \quad y_s^{new} = Q.\gamma.\delta_y, \quad z_s^{new} = Q.\gamma.\delta_z
\end{aligned}
\tag{6}
$$

where Q is the calculated similarity between the current frame described in the world coordinates by x_{curr}, y_{curr} and z_{curr} and the reference frame described by x_{ref}, y_{ref} and z_{ref}. Additionally, the correction along different axes is denoted as δ_x, δ_y and δ_z; and α is the impact of the total number of closed loops on how the $(N-K)^{th}$ pose all the way to the N^{th} pose are updated. The impact of α on the updated frames can be seen in Fig. 2, once the number of found loops increases, the previous frames are less impacted by the update rule in Equ. 6. The term γ is similar to the discount factor in reinforcement learning [23], where the impact of previous actions are weighted by this discount factor so they would not be neglected in the calculations. The same principal holds here and as the poses are calculated, the accumulated drift increases. Therefore, the impact of corrections in Equ. 6 are higher on later frames than previous ones, given higher value of γ.

Once a loop is closed, the index of the last frame at the time of loop closure is stored as the $last\ pinned\ frame$ and the algorithm allows another loop closure to be performed if the difference between the current frame index and the $last\ pinned\ frame$ is higher than a threshold called $inter\ frame\ pause$. This design parameter further reduces the computation costs in numerous scenarios, e.g., when the intelligent system is driving down a long and formerly traversed road which already triggered loop closure.

Furthermore, in Table I, a list of the hyper parameters with their functionalities are provided which enable SNAPS for SLAM to be customizable when needed. The results of SNAPS for SLAM can be seen in Fig. 3: the proposed method

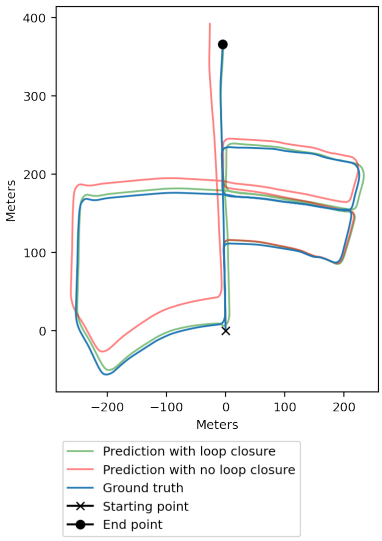

Fig. 3. Impact of loop closure in the consistency of the generated map

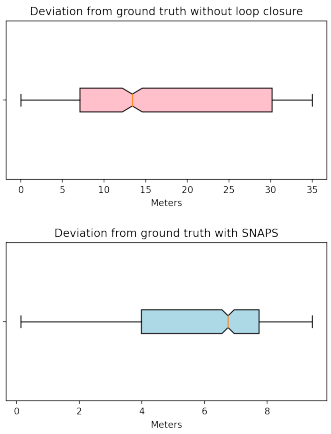

Fig. 4. Impact of loop closure in the accuracy of the generated map

significantly reduced the impact of the accumulated drift in visual odometry via SNAPS. Despite the accuracy of the proposed method, SNAPS for SLAM can run in real-time and does not require offline optimizations to provide a consistent map. More information about the runtime latency of SNAPS for SLAM are provided in section III-D. Additionally, the proposed method reduces the mean of deviations from the ground truth by 65%, decreasing it from 16.79 meters (without loop closure) to 5.94 meters (Fig. 4).

D. Runtime latency statistics

The proposed method is developed and test on MacBook Pro with 2.0 GHz quad-core 10th-generation Intel Core i5 CPU and 16GB of 3733MHz LPDDR4X RAM without any use of GPUs.

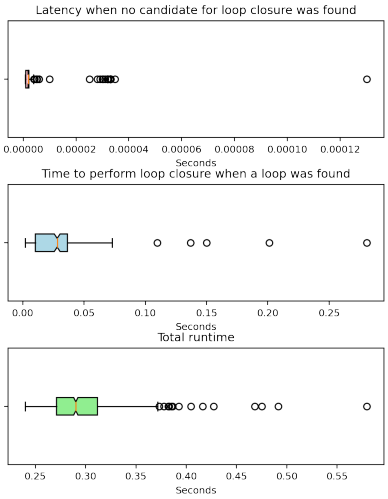

Fig. 5. SNAPS for SLAM runtime latency

Moreover, the box-plot for the computation time when the loop closure algorithm does not find candidate frames, when the candidate frame is found and lastly the total computation time of visual odometry and loop closure are depicted in Fig. 5. These have the mean values of $2.94e-6$, 0.0270 and 0.295 seconds respectively implicating the efficiency of our proposed method.

IV. Discussion and Conclusion

In this paper an interpretable and efficient model for performing loop closure in unexplored environments was introduced. The proposed method does not require any training dataset for either generating bags-of-word nor training computationally expensive deep neural networks for detecting loops. The allocated design parameters in the introduced model were defined in such a way that they were not only intuitive to the human operator, but also could be used to impact the behavior of SNAPS for SLAM. In addition, numerous elements were introduced in the SNAPS for SLAM to reduce the computational costs of this model which in the end helped the model reach mean runtime latency of only 0.0270 seconds for performing loop closure. The aforementioned elements include: use of thumbnails for calculating inter-frame similarity, checks done by the design parameters such as the search radius for finding candidate frames, frame difference and inter-frame pause, etc. The low runtime latency of the algorithm did not deteriorate its performance and as shown in this paper, SNAPS was able to significantly improve the consistency of the generated map from the KITTI dataset by reducing the mean deviation from the ground truth by 65%. The employed blaming factor trajectory correction ensures that the frames impacted more by drift in visual odometry are adjusted more according to the reference frame compared to the previous trajectory points. For the future work of SNAPS for SLAM it has been considered to employ a probabilistic

Design parameter	Details	Hints and/or tuning impact
Radius	The radius of the area around the current frame checked for loop closure	When decreased, reduces the computations but increases the chance of missing a loop closure candidate
Frame difference	The index difference between the current and the candidate reference frame for loop closure	By increasing it, more frames with indices close to the last frame are ignored for loop closure
Discount factor length	Number of datapoints in the trajectory to be updated given the corrections δ_x, δ_y and δ_z	Depending on the prior belief on visual odometry drift, increasing this parameter reduces the impact of this drift
Minimum similarity	The minimum required inter-thumbnail similarity for loop closure	Should be increased when the scene has repetitive surface materials
Inter-frame pause	The required index difference between the current candidate for loop closure and the last one	Should be increased, in case that the initial loop closures are not reliable
Attempts	The number of candidates considered for loop closure	When decreased, reduces the computations but increases the chance of missing a loop closure

TABLE I
DESIGN PARAMETERS AND THEIR TUNING GUIDES

model to adapt the parameters such as α and the constant values in the sigmoid function given the prior knowledge about the movement model and the presumed noise in the sensor readings. Despite the acceptable performance of the proposed method in the presence of pedestrians, bicycles and other cars in the scene, another potential further development of SNAPS for SLAM can be the fusion with deep neural networks for removing the temporary and dynamic objects in the scene for loop closure in heavy traffic and crowded scenarios. In fact, highly dynamic environments can be very challenging for conventional vSLAM algorithms, including SNAPS for SLAM, as these environments contain moving elements which can cause the visual odometry or the loop closure sub-modules to have erroneous predictions. Therefore, these scenarios require extensive study and further validation steps, making them another potential future study case for the improvement of the proposed method.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *IEEE robotics & automation magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[3] C. Stachniss, J. J. Leonard, and S. Thrun, "Simultaneous localization and mapping," in *Springer Handbook of Robotics*. Springer, 2016, pp. 1153–1176.

[4] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The slam problem: a survey," *Artificial Intelligence Research and Development*, pp. 363–371, 2008.

[5] Y. Xia, J. Li, L. Qi, H. Yu, and J. Dong, "An evaluation of deep learning in loop closure detection for visual slam," in *2017 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*. IEEE, 2017, pp. 85–91.

[6] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[7] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European conference on computer vision*. Springer, 2010, pp. 143–156.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[9] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[11] S. Arshad and G.-W. Kim, "Role of deep learning in loop closure detection for visual and lidar slam: A survey," *Sensors*, vol. 21, no. 4, p. 1243, 2021.

[12] D.-W. Shin and Y.-S. Ho, "Loop closure detection in simultaneous localization and mapping using learning based local patch descriptor," *Electronic Imaging*, vol. 2018, no. 17, pp. 284–1, 2018.

[13] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello, and W. Burgard, "Robust visual slam across seasons," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2529–2535.

[14] X. Gao and T. Zhang, "Loop closure detection for visual slam systems using deep neural networks," in *2015 34th Chinese control conference (CCC)*. IEEE, 2015, pp. 5851–5856.

[15] X. Zhang, Y. Su, and X. Zhu, "Loop closure detection for visual slam systems using convolutional neural network," in *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE, 2017, pp. 1–6.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[19] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020.

[20] J. Djolonga, J. Yung, M. Tschannen, R. Romijnders, L. Beyer, A. Kolesnikov, J. Puigcerver, M. Minderer, A. D'Amour, D. Moldovan *et al.*, "On robustness and transferability of convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 458–16 468.

[21] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 1643–1649.

[22] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE signal processing letters*, vol. 9, no. 3, pp. 81–84, 2002.

[23] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, vol. 8, no. 3, pp. 293–321, 1992.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[25] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, pp. 1–10, 2008.

[26] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[27] R. A. Hamzah and H. Ibrahim, "Literature survey on stereo vision disparity map algorithms," *Journal of Sensors*, vol. 2016, 2016.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.