
Proceedings Paper:
Zhao, C., Sun, L., Krajnik, T. et al. (2021) Monocular teach-and-repeat navigation using a deep steering network with scale estimation. In: Proceedings of 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 27 Sep - 01 Oct 2021, Virtual conference (Prague, Czech Republic). IEEE, pp. 2613-2619. ISBN: 9781665417150. ISSN: 2153-0858. EISSN: 2153-0866.

https://doi.org/10.1109/IROS51168.2021.9635912

Monocular Teach-and-Repeat Navigation using a Deep Steering Network with Scale Estimation

Cheng Zhao[1], Li Sun[2], Tomáš Krajník[3], Tom Duckett[4] and Zhi Yan[5*]

***Abstract*— This paper proposes a novel monocular teach-and-repeat navigation system with the capability of scale awareness, i.e. the absolute distance between observation and goal images. It decomposes the navigation task into a sequence of visual servoing sub-tasks to approach consecutive goal/node images in a topological map. To be specific, a novel hybrid model, named deep steering network is proposed to infer the navigation primitives according to the learned local feature and scale for each visual servoing sub-task. A novel architecture, Scale-Transformer, is developed to estimate the absolute scale between the observation and goal image pair from a set of matched deep representations to assist repeating navigation. The experiments demonstrate that our scale-aware teach-and-repeat method achieves satisfying navigation accuracy, and converges faster than the monocular methods without scale correction given an inaccurate initial pose. The proposed network is integrated into an onboard system deployed on a real robot to achieve real-time navigation in a real environment.**

We release this research as an open-source project to contribute to the robot visual navigation research community: `https://github.com/dachengxiaocheng/TRN-Transformer.git` A demonstration video can be found online: `https://youtu.be/fKUptTPOGEU`

I. Introduction

Vision-based robot navigation has excellent potential for a wide range of applications. Monocular-based navigation benefits from low-cost, light-weight hardware, which can be mass produced and quickly deployed. One of the challenges in achieving long-distance monocular navigation in large-scale areas is caused by the effect known as the scale drift, which makes monocular mapping of large areas complicated. This effect originates from the fact that cameras do not provide any direct information on the metric scale of the perceived scene or to determine the real-world distance between two images captured from different positions.

Recent progress in Simultaneous Localisation and Mapping (SLAM) focuses on improving visual odometry and building accurate metric maps. The state-of-the-art (SOTA) monocular SLAM methods [1], [2] use scale propagation to estimate the scale and attempt to mitigate the scale drift through loop closure. Some research employs stereo cameras or range sensors for mapping and monocular cameras for localisation. Moreover, in VINS-Mono [3], an inertial sensor is leveraged to estimate the metric scale using IMU pre-integration. To achieve mapping of large areas, these methods require expensive sensors, such as lidar-scanners, delicately calibrated stereo cameras, or precise camera-IMU sensors.

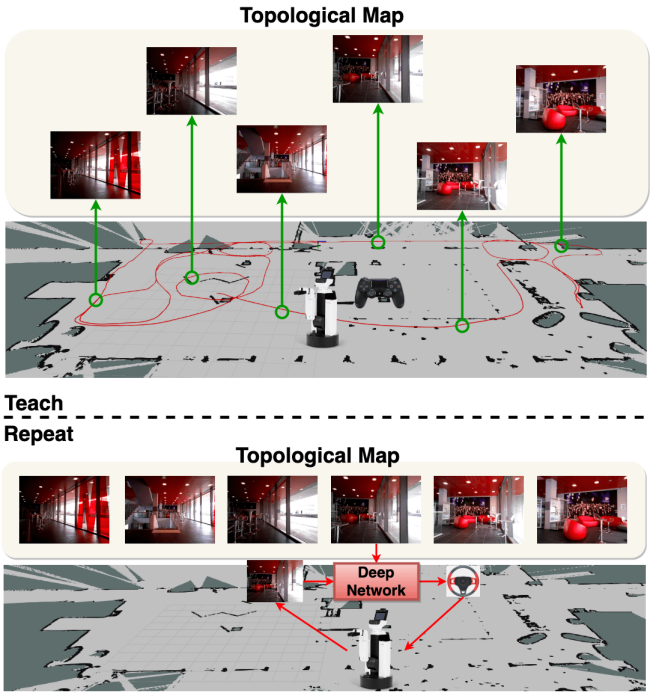

Fig. 1: Overview of monocular teach-and-repeat navigation using a topological map. Please note that the 2D grid map is only used to show the navigation environment and is not required during teach-and-repeat navigation.

Emerging approaches for learning-based visual odometry (VO) can estimate the ego-motion with absolute scale using data-driven methods [4], [5], [6], or use monocular depth estimation to recover the scale [7], [8], [9], making monocular SLAM scalable. Data-driven monocular VO approaches can be grouped into supervised [4], [5], [6] and self-supervised [7], [10] approaches. The former formulate VO as a regression problem using the powerful non-linearity of deep neural networks. The latter usually learn the ego-motion and depth simultaneously, by integrating geometric constraints into the loss function to achieve self-supervised learning.

However, navigation does not require geometrically consistent maps, and several vision-based methods do not rely on SLAM-built 3D maps of the entire environment or metric-based localisation in a global coordination frame. In particular, visual teach-and-repeat (VT&R) is a paradigm where the robot is first driven via teleoperation and can then repeat the trajectory by applying corrections from visual or

[1]Department of Engineering Science, University of Oxford, UK
[2]Visual Computing Group, University of Sheffield, UK
[3]AI Center, Czech Technical University in Prague, Czechia
[4]Lincoln Centre for Autonomous Systems, University of Lincoln, UK
[5]CIAD UMR7533, Univ. Bourgogne Franche-Comté, UTBM, France
*Corresponding Author: `zhi.yan@utbm.fr`

location residuals. In [11], [12], only localisation in local maps is required for large-scale VT&R. Topological map-based navigation is proposed in [13], [14], [15], where the local map is further simplified as consecutive images (key-frames). It is worth noting that the same navigation principle has recently been deployed on unmanned aerial vehicles [16].

There are multiple ways to navigate using a topological map. The navigation task can be formulated as a sequence of visual servoing problems [13], where the learned robot motion can be repeated through approximating consecutive goal images tied to a topological map. In the research of [13], [14], [17], [18], the visual path (topological map) is represented as a set of images captured in the teach phase. In the repeat phase, visual servoing is used to traverse through all image nodes in the map. The research in [15], [19] uses a visual offset obtained by histogram voting rather than estimating the relative pose to control the heading of the robot in the repeat phase. A proof of the convergence between teaching and repeating trajectories is provided in [15], [18], showing that the robot trajectory eventually converges to the taught path even if localisation is not performed explicitly and the scale is unknown.

Recent research [20], [21], [22] achieve path following style navigation using deep neural network. A visual memory mechanism in [20] is presented to perform path following and homing. There are two joint networks inside: one is for path abstraction generation, and another one is for acting to retrace the path. A zero-shot visual imitation mechanism in [21] is proposed to alleviate the strong supervision of expert actions to learn to imitate navigation. A deep visual model predictive control-policy learning method is proposed in [22] to perform visual navigation while avoiding collisions with unseen objects on the navigation path using a sequence of omnidirectional images. These methods achieve impressive performance in the simulation environment or small office environment. However, they don't demonstrate the ability to achieve a relatively long distance path following navigation in the real environment.

Thus monocular SLAM is not strictly necessary for robot navigation. On the other hand, bio-inspired topological representations are naturally scalable for large-scale navigation problems [23]. Our intuition is to combine scalable topological map-based localisation with deep-learned local features and scale, in order to decompose the navigation task into a series of visual servoing subtasks within each topological node, as shown in Fig. 1. The direct scale estimation obtained by the proposed deep network improves the robustness, scalability, accuracy and convergence rate of monocular teach-and-repeat navigation compared to methods that are scale-unaware. In summary, our method includes the following new features:

- A novel hybrid model, namely a deep steering network, is proposed to infer the navigation primitives, i.e. a sequence of linear and angular velocities to approach the nodes in a topological map. The robot angular velocity is inferred from matching of learned local features followed by histogram voting. The robot linear velocity is adjusted by applying absolute scale estimation on the odometry monitor to select the correct topological node map.
- A novel architecture, named Scale-Transformer, is devised to learn absolute scale between the observation and goal image pair from the matched deep representations. It can extract co-contextual information from a set of individual representations using a self-attention mechanism to enhance the feature distinctiveness.
- The deep steering model is integrated into a monocular teach-and-repeat navigation system, achieving scalable navigation using a topological map in real-time. Benefiting from the learned geometric priors, our approach can effectively eliminate the offsets derived from an inaccurate initial pose.

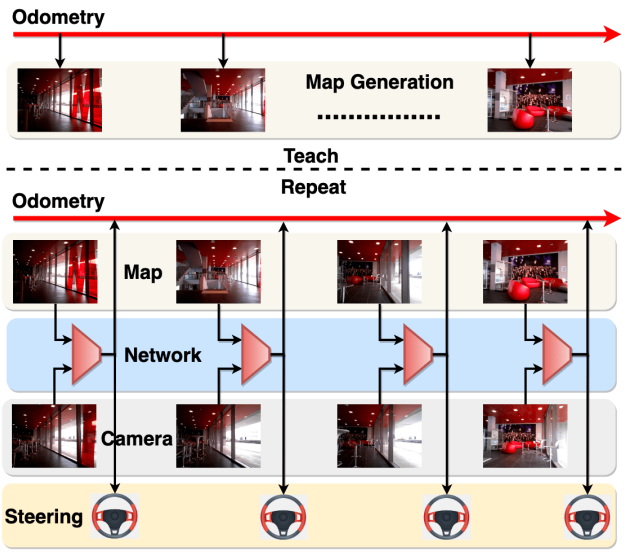

Fig. 2: Overview of the two processes: teaching (mapping) and repeating (navigation).

II. Monocular Teach and Repeat Navigation

A. Problem Formulation and Challenges

Monocular teach-and-repeat navigation includes separate phases for teaching and repeating, as shown in Fig. 2. The robot is first driven manually using a joystick while some representation of its experience is created from the on-board sensory data. The goal is to repeat the same path autonomously using this taught experience.

Most other approaches for teach-and-repeat navigation are based on localisation [11], while we use the visual servoing formulation. Given a goal image (i.e. a topological node in the map), a motion primitive should be inferred to minimise the error between the goal and on-board camera images. Instead of calculating the Jacobians, as widely used in visual servoing, we simply need to estimate the visual offset with respect to the horizontal direction of the camera (i.e., y axis) because the robot navigation mission can be simplified as a 3DoF problem. Consequently, the robot can adjust its heading according to the goal image. Once a goal image has been approached, the navigator retrieves a new goal image

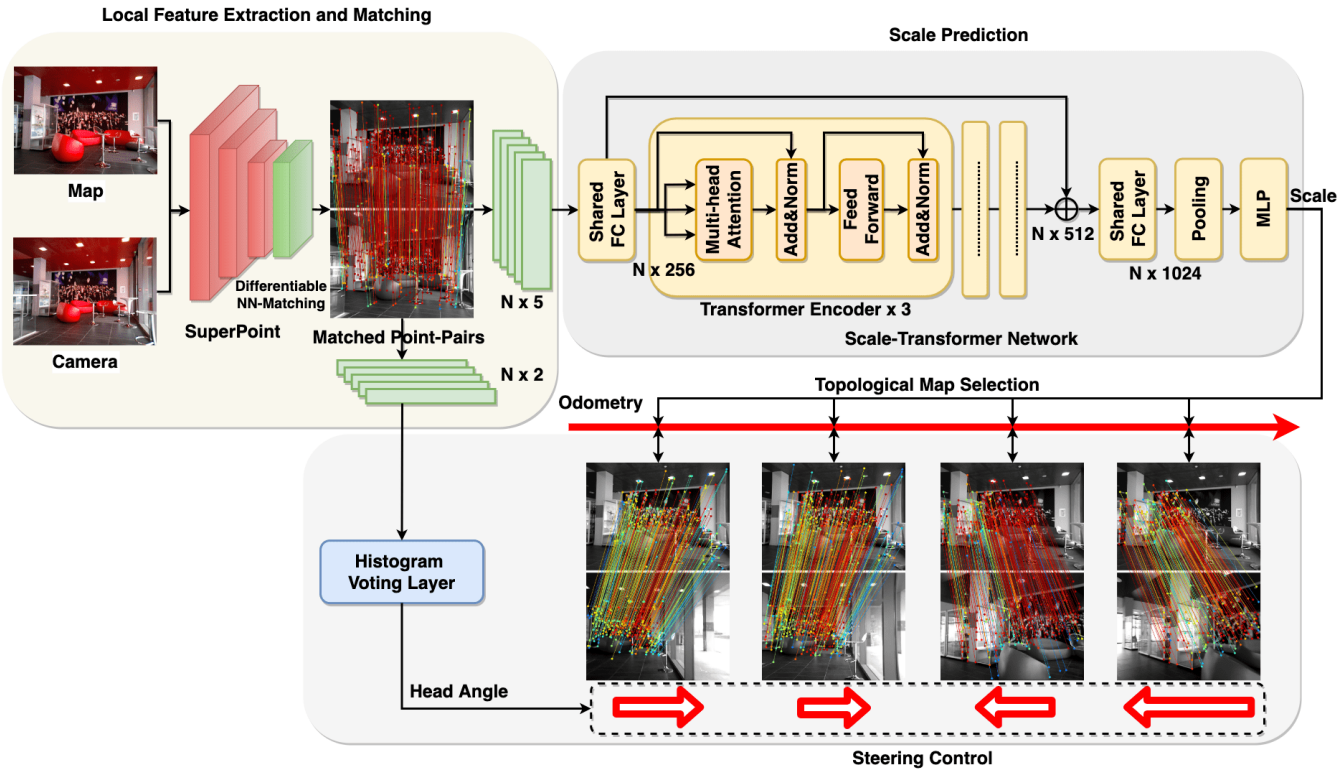

Fig. 3: Deep Steering Network Architecture

(i.e., next keyframe image) and this process is executed repeatedly until the destination image is approached.

However, monocular-based navigation faces several challenges. Firstly, the translational error cannot be estimated from two paired images due to the unknown absolute scale factor. In turn, the motion primitive on the linear direction is hard to estimate. To estimate the linear velocity, we first retrieve recorded velocities from the teaching experience (also known as the 'path profile' [18]) as a prior to initialise the motion. Then, we adjust it using our scale estimator in the deep steering network. As a result, the robot is able to estimate precise primitives (including both linear and angular velocities) to converge to the goal image. Therefore, unlike [15] and [18], our framework can efficiently correct tangential errors to the taught path. We demonstrate in the experiments that this results in faster convergence to the taught path, thereby improving the robustness and accuracy of the system.

B. Teach Phase (Mapping)

In this paper, the learned experience is represented in the form of a topological map. The topological map consists of keyframe images and the 'path profile'. To be more specific, we save the keyframe images at fixed distance intervals[1] according to the integrated translational distance provided by the wheel-encoder odometry of the robot. We further use keyframe images as submaps to simplify the mapping process and eliminate the risk of corrupted mapping for long-distance teach-and-repeat. Another benefit of topological mapping is the efficiency of running on devices of low-computational capacity. The other essential component of the map is the 'path profile' which records the joystick commands, e.g. accelerating at $2m$, turn right at $21.2m$, and stop at $52.3m$, etc. The joystick commands can be translated to continuous linear and angular velocities to represent the dynamics of the robot. The aim of using a 'path profile' is to provide the robot with a prior on movement during the repeat phase, and the traversal distance can also be used to associate corresponding map images during navigation.

[1] $0.2m$ in our experiments.

C. Repeat Phase (Navigation)

The key component of our navigation method is a deep 'steering' model which can infer the robot actions to approach goal images. The input to the steering network comprises two images, i.e. goal/map image and observation/-camera image, and they will go through the computational graph to obtain the rotational and translational errors. The rotational error is used to infer the angular velocity and the translational error is applied to the current odometry for correct goal image selection. In practice, the goal images can always be precisely located and the primitives can be assembled by the linear velocity from the 'path profile' and the angular velocity can be converted linearly from the rotational error. As a result, in each topological node, the goal image will be approximated and approached, and this process will be executed until the robot reaches the destination node. More details of the steering network can be found in the next section.

III. Deep Steering Model

The key component inside our monocular teach-and-repeat navigation system is a hybrid model, namely a deep steering network. As shown in Fig. 3, the deep steering network consists of three main components: 1) learning local feature extraction and matching, 2) Scale-Transformer, and 3) inference of navigation primitives.

A. Learning Local Feature Extraction and Matching

Given the map image I_M and camera image I_C data-pair, we feed them into a SuperPoint [24] network to extract the interest points p and descriptors d. SuperPoint is a self-supervised network consisting of a VGG-style feature encoder followed by a detector branch and a descriptor branch. We rewrite the nearest neighbor matching (NN-Matching) algorithm as a new differentiable NN-Matching layer following SuperPoint in order to achieve end-to-end trainable manner. According to the descriptor distance calculation, differentiable NN-Matching provides the matched interest point pair (p_M, p_C) with the corresponding matched descriptor distance d_{MC}.

In this paper, we choose SuperPoint with differentiable NN-Matching for the local feature extraction and matching due to its run time performance. Other more powerful methods such as SuperGlue [25] are not fast enough to be deployed in our system because of the real-time performance requirement for robot navigation in real environments.

B. Scale-Transformer for Scale Estimation

We propose a novel architecture, named Scale-Transformer, to learn scale from a set of irregular matched correspondence representations. As shown in Fig.3, the network includes 3 Transformer encoder stacks in series, 2 shared linear stacks at the head and bottom with a shortcut skip connection, and a Multi-Layer Perceptron (MLP) at the end.

Given a set of correspondences (p_M, p_C, d_{MC}), we firstly utilise a linear stack f to obtain the high-dimensional feature representation $\mathcal{F}$,

$$\mathcal{F} = f(p_M \oplus (p_M - p_C) \oplus d_{MC}), \tag{1}$$

where $\oplus$ refers to the concatenation. However, this feature representation $\mathcal{F}$ suffers from ambiguities due to a lack of contextual cues. There are no interactions between these individual representations learning from an irregular regulated data structure. Inspired by the success of Transformer [26] in natural language processing, we treat the set of individual representations $\mathcal{F}$ as a "long sentence" of feature representations. Thus, we employ a residual transformer encoder to aggregate the contextual cues for one representation from all the other representations to enhance feature distinctiveness.

The Transformer encoder consists of a series of sub-architectures including multi-head self-attention (MHSA), feed-forward network (FFN) and layer normalization (LN), which can be stacked on top of each other multiple times. The Transformer encoder extracts a co-contextual message $Attn$ captured by the MHSA mechanism,

$$Attn([Q_i, K_i, V_i]) = \text{concat}([\text{softmax}(\frac{Q_i \cdot K_i^T}{\sqrt{d_k}})V_i]), \tag{2}$$

where d_k denotes the dimension of queries and Q_i, K_i, V_i stand for the ith head of queries, keys, and values of each feature representation $\mathcal{F}$, respectively. We adopt four-head attention, $i \in [1, 2, 3, 4]$ in our implementation. The MHSA mechanism automatically builds connections between a current feature representation and the other salient interesting feature representations.

The attentional representation $\mathcal{A}$ can be obtained as,

$$\mathcal{A}' = LN(\mathcal{F} + Attn), \tag{3}$$

$$\mathcal{A} = LN(FFN(\mathcal{A}') + \mathcal{A}'). \tag{4}$$

After going through the 3 Transformer encoder stacks in series, the individual representation $\mathcal{F}$ of each correspondence is upgraded to attentional representation $\mathcal{A}$.

Finally, a MLP regression is used to predict the scale $\mathcal{S}$ between the map image I_M and camera image I_C,

$$\mathcal{S} = MLP(f^{'}(\mathcal{F} \oplus \mathcal{A})), \tag{5}$$

where $\oplus$ refers to the concatenation and $f^{'}$ is another linear stack. Here, the scale $\mathcal{S} = (\mathcal{S}_x, \mathcal{S}_y, \mathcal{S}_z)$ refers to a three-dimensional translation between the map image I_M and camera image I_C. The coordinate system in this paper is defined as, x : right (horizontal), y : down (vertical), z : forward (horizontal). The ℓ_2 loss function is employed during the network training,

$$\mathcal{L}(I_M, I_C) = \| \mathcal{S} - \mathcal{S}_{MCL} \|_2 + \lambda \cdot \|W\|_2, \tag{6}$$

where W are the trainable weights and λ is a scale factor. $\mathcal{S}_{MCL}$ is the "ground truth" scale obtained by the 2D laser-based Monte-Carlo Localization (MCL) with a previously built 2D grip map. Please note, the 2D laser and 2D grid map are only used to provide ground truth for neural network training and they are not required during the test/repeat navigation. These ground truth can also be obtained by the laser/lidar, Wicon system indoors or GPS-RTK outdoors.

C. Navigation Primitives Inference

As formulated in Section II, we decompose the navigation task into visual servoing problems within each topological node. To drive the robot to approach each goal image, a primitive consisting of a sequence of linear and angular velocities needs to be estimated. As our Scale-Transformer model can estimate the translation $(\mathcal{S}_x, \mathcal{S}_y, \mathcal{S}_z)$ between observation/camera image and goal/map image with absolute scale, a natural idea is to use the translation in the navigation direction and its perpendicular (i.e. $\mathcal{S}_z$, $\mathcal{S}_x$) to correct the motion priors, i.e. recorded teaching actions. However, in practice, this does not generate a robust policy since the unreliable predictions may drive the robot to an unknown state.

To improve the robustness of autonomous navigation, a

histogram voting layer is applied to the pixel visual offset to infer the angular velocity. Specifically, we calculate the visual offset $\Delta x, \Delta y$ in x and y directions separately for all deep feature matches. Then Δy is capped at a maximum of 10 to filter incorrect matches, and Δx of the filtered matches will be used to build the histogram. The robust statistical estimate of Δx can be obtained by calculating the average of the largest bin (inliers) in the histogram. The final angular velocity can be obtained by adding a compensation vel_{gain} to the recorded action, where vel_{gain} is inferred from the mean value of inliers of Δx with multiplication by a constant factor[2].

Instead of adjusting the linear velocities using the estimated scale directly, we use the recorded action, but apply $\mathcal{S}$ on the odometry monitor to select the correct goal/map image. We add a momentum term to $\mathcal{S}$ over time Δt to smooth this correction,

$$\mathcal{S}_t = (1-\lambda)\mathcal{S}_t + \frac{\lambda}{n_{\Delta t}} \int \mathcal{S}_t dt, \tag{7}$$

where $n_{\Delta t}$ is number of estimates within Δt, with the assumption that observation and actions within Δt are non-i.i.d. We set the coefficient $\lambda = 0.9$ and $\Delta t = 2s$ in our paper. By this means, the robot will be driven in a proper linear velocity with minimisation of the hazard of anomalous acceleration/deceleration, and importantly the robot can always localise correctly in the topological map.

IV. Experiment

A. Robot Platform and Data Collection

In this paper, we employ a Toyota Human Support Robot (HSR) [27] equipped with an ASUS Xtion PRO LIVE camera and a Hokuyo UST-20LX laser scanner for data collection and system deployment. The network is deployed on a laptop (carried by the robot on its back) with a NVIDIA Geoforce GTX 1060 GPU. The whole navigation system is fully implemented into the ROS [28] framework using C++ and Python mixed coding. The runtime performance of the deep steering network varies within 20-30Hz using 480×320 images, depending on the type of power supply used.

We recorded two long ROS bags in large hall of around $2500m^2$ at the UTBM building following different routes. The RGB images were captured by the Xtion camera, and the corresponding global poses were recorded by the laser scanner via MCL localization on the pre-built 2D grid map. After data calibration and reprocessing, we obtain a long sequence consisting of 20940 synchronized image-pose data for network training, and a long sequence consisting of 19247 synchronized image-pose data for network testing.

B. Network Hyper-Parameters and Training

Regarding the network hyper-parameters, the two shared linear stacks consist of linear layers with hidden variables 256 and 1024, respectively, followed by a Batch Normalisation layer and ReLU activation layer. The three transformer encoders are set with the same hyper-parameters: the dimensions of the input and output are 256, the dimension of the feed-forward layer is 512, the number of heads of self-attention layer is 4, and the dropout is 0.1. There are two linear layers with hidden variables 512, 128 followed by a Batch Normalisation layer and ReLU activation layer, one linear layer with hidden variables 3 in the MLP.

The network is trained on the training sequence consisting of 20940 synchronized image-pose data. During training, we feed a batch of image pairs (I_i, I_j), where j is randomly selected from $[i-50, i+50]$, extracted from the training sequence to the network. The corresponding relative global poses are used as ground truth. The network is trained for 30 epochs with a batch size of 32. The input images are cropped to 480×320. A step learning policy is employed, and the learning rate decay is fixed to 0.1 applied on every 5 epochs. The initial learning rate is 1e-3 and the momentum is fixed to 0.9. The pre-trained SuperPoint model is used to initialize the weights of the local feature sub-network. In order to increase the robustness of training, gradient clipping is utilised during training. The network is implemented via Pytorch and is trained on an NVIDIA Titan X GPU accelerated by CUDA and cuDNN.

C. Scale Evaluation

We evaluate the scale prediction on the test sequence consisting of 19247 synchronized image-pose data. We feed an image pair (I_i, I_j), where j is randomly selected from $[i-50, i+50]$, into the network to obtain the predicted translation. The mean and median of the ℓ_2 norm distance offset between the predicted translation and the ground truth translation are used as evaluation metrics.

We provide the results of four baseline methods as a comparison. The first and second baselines are classic supervised deep learning-based VO [4][29] and self-supervised deep learning-based VO [10][30], which focus on solving the scale ambiguity between two monocular images. For the third baseline, we replace the Scale-Transformer with a PointNet [31] to regress the scale from a set of individual matched correspondence representations. For the fourth baseline, we add a RANSAC model after SuperPoint and differentiable NN-Matching. The feature-matching-based RANSAC can only predict a norm vector to describe the directions of the translation. We modify the last linear layer in the Scale-Transformer to predict a one-dimensional scale factor from a set of matched correspondence representations. This scale factor is further multiplied with the norm vector to obtain the predicted translation.

The evaluation results are shown in Table I. The proposed method achieves $0.054m$ for mean translation error and $0.037m$ for median translation error. We can see that the performance of Scale-Transformer outperforms both supervised deep learning-based VO [4][29] and self-supervised deep learning-based VO [10][30]. Most importantly, most existing learning VO methods, especially one of which [10][30] including depth perdition using a decoder style network architecture, cannot satisfy the real-time performance re-

[2]In this paper, a factor of 0.02 is used to re-scale the visual offset in pixels to the angular velocity (yaw) in radian/s.

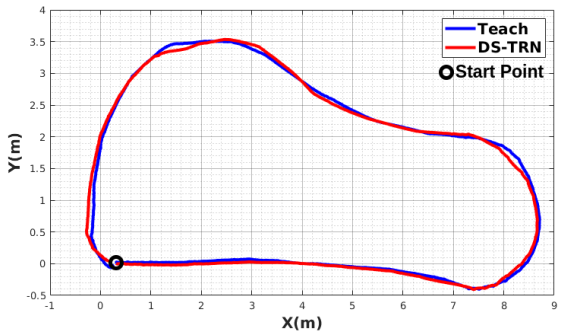

Fig. 4: Teaching and repeating trajectories of DS-TRN.

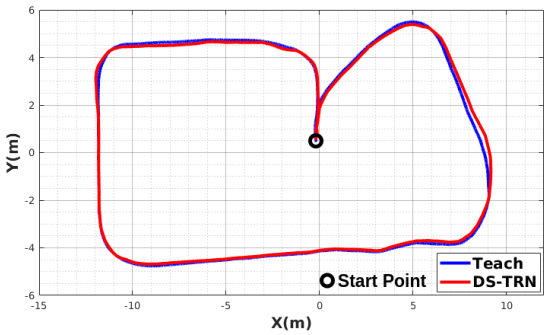

Fig. 5: Teaching and repeating trajectories of DS-TRN in a large open space indoor environment.

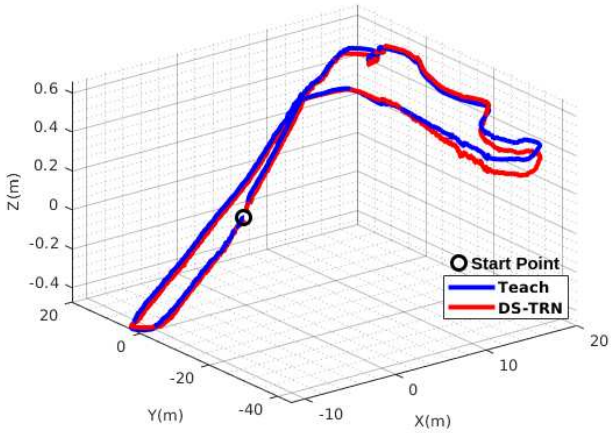

Fig. 6: Teaching and repeating trajectories of DS-TRN in an out-of-plane outdoor environment.

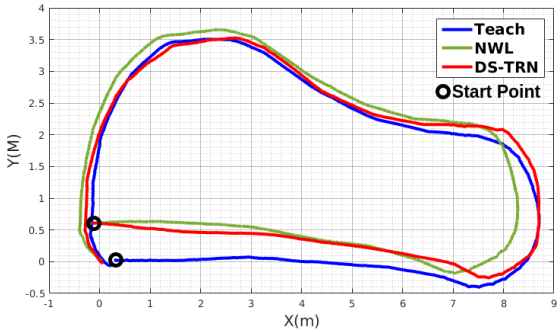

Fig. 7: Comparison of navigation trajectories for the proposed method and baseline method [18] at a start position with offset.

quirement (at least 20Hz) of navigation deployed on the mainstream robot-mounted hardware. In addition, the proposed method is more customized for the deep steering network using the learned local feature matches rather than the original RGB image.

It can also be seen that the performance of the proposed network is superior to the PointNet-based regression. The reason is that PointNet [31] ignores all the contextual cues within the deep feature matched correspondences, while the Scale-Transformer can learn the co-contextual information from these individual representations using a self-attention mechanism. Including the RANSAC model provides a slight performance improvement due to its ability to remove outliers. However, including RANSAC-based outlier removal makes the system cannot be deployed in real time, especially when the field of view between the map and camera images varies significantly during robot navigation.

Method	Mean Error	Median Error
DeepVO[4][29]	0.113 m	0.089 m
UnDeepVO[10][30]	0.172 m	0.138 m
SuperPoint+NN+PN[31]	0.104 m	0.081 m
SuperPoint+NN+ST	0.054 m	0.037 m
SuperPoint+NN+RS+ST	**0.041** m	**0.032** m

TABLE I: Comparison of scale estimation performance for the proposed method and baseline methods. NN: Differentiable NN-Matching, PN: PointNet [31], ST: Scale-Transformer, RS: RANSAC.

D. Navigation Evaluation

We also test the teach-and-repeat navigation integrated with the deep steering network on the Toyota HSR robot. The pure learning-based navigation approaches such as [20], [21], [22] mainly focus on the success rate of indoor navigation, while the proposed hybrid model is extremely difficult to be navigation failed. So the repeating navigation (path following) accuracy is more suitable to evaluate the proposed method. Similar to the localization-style navigation methods [15], [18], we also choose the localization error of path following as the evaluation criterion. The absolute trajectory error, i.e. mean ℓ_2 norm distance of positional offset, and relative rotation error, i.e. mean absolute orientation offset, between the corresponding poses in the teach and repeat trajectories are employed as evaluation metrics. The robot poses in the teach and repeat trajectories are associated by searching for the mutual nearest distance according to the odometry distance.

During the first experiment, we test the Deep Steering Teach-and-Repeat Navigation (DS-TRN) starting at the exact same position as the start point of the teaching trajectory. As shown in Table II, DS-TRN achieves $0.15m$ for absolute trajectory error and 1.0° for relative rotation error. From Fig. 4, it can be seen that the navigation trajectory of DS-

TRN follows the teaching trajectory very closely. We also test the DS-TRN in a large open space indoor environment as shown in Fig. 5. Despite some increase, the absolute trajectory error $0.17m$ and the relative rotation error $1.9°$ are still acceptable for navigation in large open spaces indoor environment. We finally test the DS-TRN in an out-of-plane outdoor environment as shown in Fig. 6. A DrRobot Jaguar 4x4 robot equipped with ZED2 camera, 3D Ouster OS1-64 lidar, Xsens MTi-G710-GNSS is used for the outdoor navigation test, and the ground through is obtained by lidar-inertial-GPS SLAM and ICP registration. We achieve absolute trajectory error $0.18m$ and the relative rotation error $2.6°$ for out-of-plane outdoor navigation.

Method	Abs. Traj. Error	Rel. Rot. Error
DS-TRN	**0.15 m**	**1.0°**
DS-TRN@L	0.17 m	1.9°
DS-TRN@O	0.18 m	2.6°

TABLE II: Navigation performance of DS-TRN at an exact start point. @L refers to the test in a large, open space indoor environment. @O refers to the test in an out-of-plane outdoor environment.

Method	Abs. Traj. Error	Rel. Rot. Error
NWL[18]	0.57 m	9.3°
DS-TRN	**0.28 m**	**3.0°**

TABLE III: Comparison of navigation performance for the proposed method and baseline method [18] at a start point with offset.

In order to increase the challenge of the navigation task, during the second experiment, we test the repeat navigation starting at a position $0.5m$ left and $0.5m$ backward from the start point of the teaching trajectory. A classic monocular teach-and-repeat navigation method, Navigation Without Localization (NWL) [18], is employed as a baseline for the performance comparison. NWL [18] uses SIFT feature as visual frontend and there is no scale estimation inside. From Table III, we can see that the navigation performance of DS-TRN is superior to that of NWL with $0.29m$ improvement for absolute trajectory error and $6.3°$ improvement for relative rotation error. The improvement mainly comes from the velocity adjustment according to selecting the correct topological node map through applying scale estimation on the odometry monitor. This helps the repeating navigation "catch up" with the teaching navigation as quickly as possible by reducing the difference caused by the starting point offsets. As shown in Fig. 7, the trajectory of DS-TRN converges faster than NWL to the taught trajectory. After the first turn (bottom right corner in Fig. 7), the trajectory of DS-TRN becomes closer to the taught trajectory than NWL. Another important reason is that SuperPoint-features provides richer position and descriptor information than SIFT features to adjust the robot's angular velocity.

We also try to deploy the more complex deep learning based navigation methods, such as [20], [21], [22] on our robot as comparison. However, we find that they do not achieve successful real-time navigation on real robot for relatively long distance navigation test at $2500m^2$ hall building or outdoor environment perhaps due to our limited onboard computing resources or different configuration of robot hardware.

More information can be found on our project website: `https://github.com/dachengxiaocheng/TRN-Transformer.git`

V. Conclusion

In this paper, we propose a novel monocular teach-and-repeat navigation method with scale estimation between observation and goal images in the topological map. The proposed deep steering network is a hybrid computational graph that encapsulates the robot heading estimation from matching the learned local features and precise topological positioning using the learned absolute scale. With the help of the learned geometric priors, our monocular navigation not only achieves satisfying navigation accuracy, but can also deal with coarse start pose initialization during autonomous navigation. For the future work, we will deploy the proposed navigation method in a large-scale outdoor environment.

VI. Acknowledgement

This work is supported by the Toyota Partner Robot joint research project (MACPOLO), EPSRC FAIR-SPACE Hub (EP/R026092/1), and EU Horizon 2020 ILIAD (No. 732737) and CSF/NRF project ToLTATempo 20-27034J.

References

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[3] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[4] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2043–2050.

[5] C. Zhao, L. Sun, P. Purkait, T. Duckett, and R. Stolkin, "Learning monocular visual odometry with dense 3d mapping from dense 3d flow," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6864–6871.

[6] C. Zhao, L. Sun, Z. Yan, G. Neumann, T. Duckett, and R. Stolkin, "Learning kalman network: A deep monocular visual odometry for on-road driving," *Robotics and Autonomous Systems*, vol. 121, p. 103234, 2019.

[7] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[8] W. N. Greene and N. Roy, "Metrically-scaled monocular slam using learned scale factors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 43–50.

[9] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992.

[10] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7286–7291.

[11] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.

[12] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.

[13] G. Blanc, Y. Mezouar, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 3354–3359.

[14] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *Proceedings of IEEE International conference on Robotics and Automation*, vol. 1. IEEE, 1996, pp. 83–88.

[15] T. Krajník, J. Faigl, V. Vonásek, K. Košnar, M. Kulich, and L. Přeučil, "Simple yet stable bearing-only navigation," *Journal of Field Robotics*, vol. 27, no. 5, pp. 511–533, 2010.

[16] T. Nguyen, G. K. Mann, R. G. Gosine, and A. Vardy, "Appearance-based visual-teach-and-repeat navigation technique for micro aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 217–240, 2016.

[17] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette, "Large scale vision-based navigation without an accurate global reconstruction," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[18] T. Krajník, F. Majer, L. Halodová, and T. Vintr, "Navigation without localisation: reliable teach and repeat based on the convergence theorem," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1657–1664.

[19] F. Majer, L. Halodová, T. Vintr, M. Dlouhỳ, L. Merenda, J. P. Fentanes, D. Portugal, M. Couceiro, and T. Krajník, "A versatile visual navigation system for autonomous vehicles," in *International Conference on Modelling and Simulation for Autonomous Systesm*. Springer, 2018, pp. 90–110.

[20] A. Kumar, S. Gupta, D. Fouhey, S. Levine, and J. Malik, "Visual memory for robust path following," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 31. Curran Associates, Inc., 2018.

[21] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell, "Zero-shot visual imitation," in *International Conference on Learning Representations (ICLR)*, 2018.

[22] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese, "Deep visual mpc-policy learning for navigation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3184–3191, 2019.

[23] B. Kuipers, "The spatial semantic hierarchy," *Artificial intelligence*, vol. 119, no. 1-2, pp. 191–233, 2000.

[24] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[25] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[27] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of human support robot as the research platform of a domestic mobile manipulator," *ROBOMECH Journal*, vol. 6, no. 4, 2019.

[28] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[29] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.

[30] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, no. 2, p. 4, 2017.